AD NUMBER
ADB131157
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Critical Technology; JAN 1989. Other requests shall be referred to Aviation Applied Technology Dir., AVSCOM, Fort Eustis, VA. This document contains export-controlled technical data.**
AUTHORITY
AVSCOM ltr 22 Mar 1990

USAAVSCOM TR-88-D-14B

US ARMY
AVIATION
SYSTEMS COMMAND

DYNAMIC SYSTEM COUPLER PROGRAM (DYSCO 4.1)
VOLUME II - USER'S MANUAL

Alex Berman, Shyi-Yuang Chen, Bruce Gustavson, Patricia Hurst

Kaman Aerospace Corporation
P.O. Box 2
Bloomfield, CT 06002-0002

January 1989

Final Report for Period September 1985 - May 1988

Distribution authorized to U.S. Government agencies and their contractors, critical technology, January 1989. Other requests for this document shall be referred to the Aviation Applied Technology Directorate, U.S. Army Aviation Research and Technology Activity (AVSCOM), Fort Eustis, VA 23604-5577.

WARNING

This document contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Section 2751 et seq.) or Executive Order 12470. Violators of these export laws are subject to severe criminal penalties.

Prepared for

AVIATION APPLIED TECHNOLOGY DIRECTORATE
US ARMY AVIATION RESEARCH AND TECHNOLOGY ACTIVITY (AVSCOM)
Fort Eustis, VA. 23604-5577

AVIATION APPLIED TECHNOLOGY DIRECTORATE POSITION STATEMENT

This report documents the work performed to enhance the Dynamic System Coupler (DYSCO) computer program through the addition of advanced modeling capabilities. These capabilities include rotor blade damage modeling, Eigen analysis development, general time history solution development, frequency domain solution development, general modal representation of three-dimensional structures, lifting surface modal representation, landing gear, general force, linear constraints, lifting surface aerodynamics, calculation of component interface and internal loads, and a nonlinear spring and damper system. While the improvements incorporated into DYSCO, as a result of this work, increase the analytical capabilities of the program, it still has limitations in several areas. More correlation with flight test data or with similar proven analytical tools is needed to validate program results. A new or improved trim algorithm is needed to eliminate deficiencies in the current DYSCO trim algorithm. Also, DYSCO should be converted to double precision to increase the accuracy of program results.

Mr. Robert A. Lindholm of the Aeronautical Technology Division served as the project engineer for this contract.

The following notice applies to any unclassified (including originally classified and now declassified) technical reports released to "qualified U.S. contractors" under the provisions of DoD Directive 5230.25, Withholding of Unclassified Technical Data From Public Disclosure.

NOTICE TO ACCOMPANY THE DISSEMINATION OF EXPORT-CONTROLLED TECHNICAL DATA

1. Export of information contained herein, which includes, in some circumstances, release to foreign nationals within the United States, without first obtaining approval or license from the Department of State for items controlled by the International Traffic in Arms Regulations (ITAR), or the Department of Commerce for items controlled by the Export Administration Regulations (EAR), may constitute a violation of law.

2. Under 22 U.S.C. 2778 the penalty for unlawful export of items or information controlled under the ITAR is up to two years imprisonment, or a fine of $100,000, or both. Under 50 U.S.C., Appendix 2410, the penalty for unlawful export of items or information controlled under the EAR is a fine of up to $1,000,000, or five times the value of the exports, whichever is greater; or for an individual, imprisonment of up to 10 years, or a fine of up to $250,000, or both.

3. In accordance with your certification that establishes you as a "qualified U.S. Contractor", unauthorized dissemination of this information is prohibited and may result in disqualification as a qualified U.S. contractor, and may be considered in determining your eligibility for future contracts with the Department of Defense.

4. The U.S. Government assumes no liability for direct patent infringement, or contributory patent infringement or misuse of technical data.

5. The U.S. Government does not warrant the adequacy, accuracy, currency, or completeness of the technical data.

6. The U.S. Government assumes no liability for loss, damage, or injury resulting from manufacture or use for any purpose of any product, article, system, or material involving reliance upon any or all technical data furnished in response to the request for technical data.

7. If the technical data furnished by the Government will be used for commercial manufacturing or other profit potential, a license for such use may be necessary. Any payments made in support of the request for data do not include or involve any license rights.

8. A copy of this notice shall be provided with any partial or complete reproduction of these data that are provided to qualified U.S. contractors.

DESTRUCTION NOTICE

For classified documents, follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX. For unclassified, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Distribution authorized to U.S. Government agencies and their contractors, Critical Technical, January 1989. Other requests for this document shall be referred to the Aviation Applied Technology Directorate, U.S. Army Aviation Research & Technology Activity (AVSCOM), Fort Eustis, VA 23604-5577.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) R-1790	5. MONITORING ORGANIZATION REPORT NUMBER(S) USAAVSCOM TR 88-D-14B
6a. NAME OF PERFORMING ORGANIZATION KAMAN AEROSPACE CORPORATION 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AVIATION APPLIED TECHNOLOGY DIRECTORATE
6c. ADDRESS (City, State, and ZIP Code) P.O. BOX 2 BLOOMFIELD, CONNECTICUT 06002-0002	7b. ADDRESS (City, State, and ZIP Code) U.S. ARMY AVIATION RESEARCH AND TECHNOLOGY ACTIVITY (AVSCOM) FORT EUSTIS, VIRGINIA 23604-5577
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAJ02-85-C-0033
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
62209A	1L1622-09AH76	B	43EK

11. TITLE (Include Security Classification)

Dynamic System Coupler Program (DYSCO 4.1), Volume II - User's Manual

12. PERSONAL AUTHOR(S)
Alex Berman, Shyi-Yuang Chen, Bruce Gustavson, Patricia Hurst

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 9/13/85 TO 5/13/88	January 1989	354

16. SUPPLEMENTARY NOTATION

Volume II of a three-volume report.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Dynamic analysis, Component coupling, Helicopter analysis.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

DYSCO is an interactive computer program which allows a user to model the dynamic and aerodynamic behavior of rotorcraft and other aerospace structures. The program provides user-oriented modeling procedures and data base management for dynamic analyses of coupled systems of independently modeled components. Each component is a system of second-order ordinary differential equations. The user employs modules from an expandable library of algorithms to formulate and couple components and to obtain solutions of the coupled system.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL R.A. Lindholm	22b. TELEPHONE (Include Area Code) (804) 878-3773 — 22c. OFFICE SYMBOL SAVRT-TY-ATA

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1.0 GENERAL OVERVIEW

DYSCO is a program which provides general capabilities in the specific domain of dynamic and aerodynamic analysis. It is based on an approach that couples the equations of motion of independent components and force algorithms into a "model" which may be processed by various solution procedures.

Primary features of DYSCO are: interactive operation with optional batch mode capabilities; validated input and stored data sets; convenient editing of data and models; nonlinear equations; arbitrary types of generalized coordinates; highly automated component coupling; and expandable technology library of component representations, forcing equations, and solution algorithms.

This report provides all the information necessary to make use of the program and to install new technology modules.

1.1 DYSCO LIBRARIES

The operation of the program may be described by an examination of the system architecture from the perspective of the DYSCO libraries, as shown on Figure 1.

The Executive System controls all operations involving the three separate libraries. The functions and uses of the libraries are described in the following paragraphs.

1.1.1 Technology Module Library. All specific analysis technology is included in this area. Each technology module consists of several FORTRAN subroutines. Details may be found in Section 4. For use of the program, however, it is only necessary to understand the basic functions of the three types of technology modules.

"Component modules," named in the form C---, perform the following principal functions: accept input data to be stored on user files (see paragraph

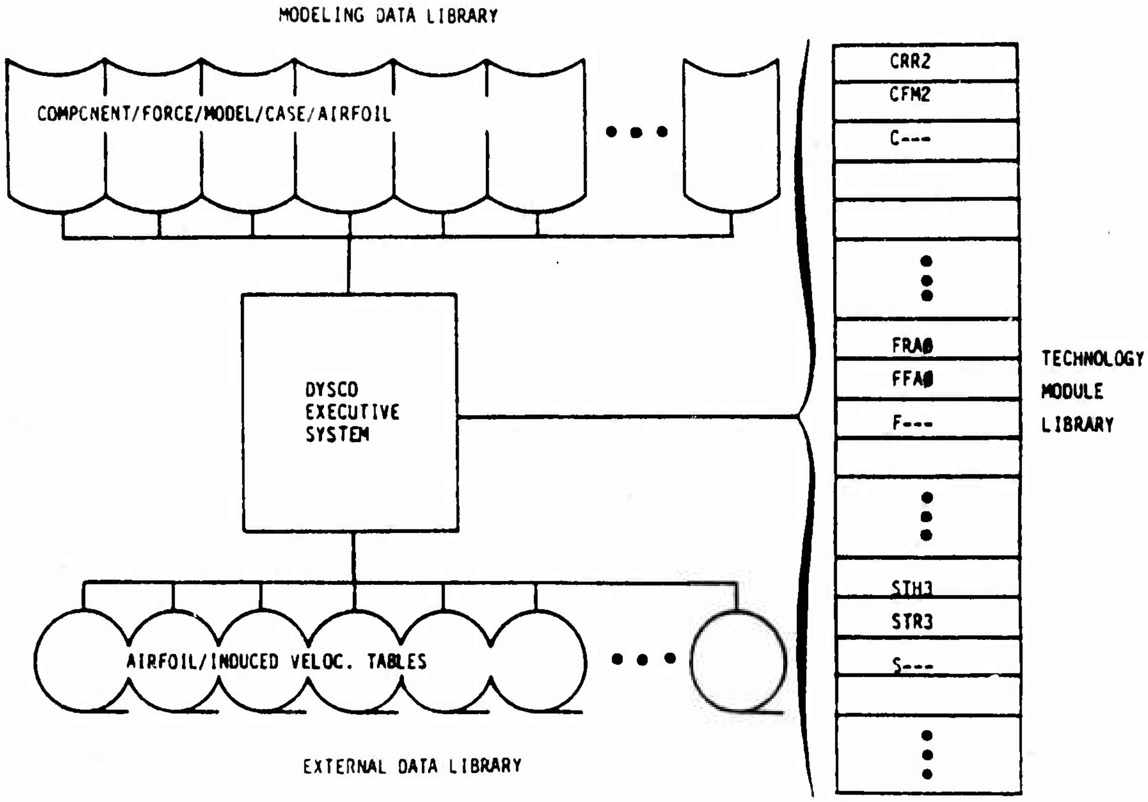


Figure 1. DYSCO Libraries.

1.1.2); define degrees of freedom in a particular application; compute matrix coefficients of second-order differential equations of component; and establish interfaces to "Force modules" as necessary (see next paragraph).

"Force modules," named in the form F---, perform the following principal functions: accept input data to be stored on user files (see paragraph 1.1.2) and compute forces applied to a particular component based on information supplied by a component module during a solution procedure.

"Solution modules," named in the form S---, perform the following functions: accept input; carry out numerical solutions of specified "model" (combination of components and forces); and provide output data.

Samples of technology modules are:

CRR2	Rigid blade, hinged helicopter rotor
CSF1	Structure, finite element
CFM2	Fuselage, modal representation
FRAØ	Rotor aerodynamic force
FSS1	Sinusoidal shaker force
STH3	Time history solution
SEA4	Eigenanalysis solution

Details of all installed technology modules are given in Section 3. New technology modules may be added to the library, as described in Section 4.

1.1.2 <u>Modeling Data Library</u>. All data associated with specific applications of component and force technology modules and definitions of models are stored in this library. Data is originally input as one of the functions of the C--- and F--- technology modules.

The library consists of any number of user files; however, only up to four may be attached to the program at one time. In addition, a temporary file is

always available which is initialized at the start of each execution of the program.

Each data unit on the files is uniquely identified by a data set/data member (ds/dm) name. A data unit (ds/dm) may contain a complete set of input for a particular technology module. The dm name is automatically supplied by the Executive during input and identifies how the data is to be used. For data input through CSF1, for example, the dm is "CSF1." The ds name is arbitrary and is provided by the user during the input of the data.

Samples of unique data units (ds/dm) could be:

AH1J3/CRR2
AH1J3/CFM2
AH1J3/MODEL
XXXX/CRR2

A complete description of usage of the user files is given in Section 2.

1.1.3 External Data Library. This library consists of up to two sequential files which may be attached to the program during any run. Its purpose is to provide another source for data. At present, airfoil tables and induced velocity tables may be accessed in this manner. Airfoil tables may be read and converted into a DYSCO format and stored on a user file with the assigned dm name, AIRFOIL.

This library, in the future, could be used to accept data from other programs, such as NASTRAN or a CFD program.

1.2 COUPLING

Each component represents a free independent structure. A model is formed by attaching the components to each other to form a system of components, i.e., a

"model." The procedure used in DYSCO to define the intercomponent attachments is described in the following paragraphs.

1.2.1 Coordinate Transformations. Each component is represented by a set of equations of the form

$$M_I \ddot{X}_I + C_I \dot{X}_I + K_I X_I = F_I + F_{IR} \tag{1}$$

where M_I, C_I, and K_I are the coefficient matrices of component I, F_I is the forcing function vector, and F_{IR} is a vector of the reaction forces at the interfaces to other components. M_I, C_I, K_I, and F_I may be nonlinear functions of the component state vector and/or time. F_{IR} is unknown. X_I is the vector of displacements of the generalized coordinates of the component.

The equations of a model (a set of coupled components) are of the form

$$M\ddot{X} + C\dot{X} + KX = F \tag{2}$$

where M, C, K, F, and X are defined as for the component.

Consider a matrix, T_I, which transforms X into X_I:

$$X_I = T_I X \tag{3}$$

If the components are physically joined, T_I is constant and it can be shown[1] that

[1]Hurty, W. C., "Dynamic Analysis of Structural Systems by Component Mode Synthesis," Technical Report 32-530, Jet Propulsion Laboratory, Pasadena, California, January 1964.

$$M = \sum_I T_I^T M_I T_I$$

$$C = \sum_I T_I^T C_I T_I$$

$$K = \sum_I T_I^T K_I T_I \qquad (4)$$

$$F = \sum_I T_I^T F_I$$

and

$$\sum_I T_I^T F_{IR} = 0$$

The requirement that T_I be constant (not a function of time) implies that components are attached rigidly in a physical sense. If two components are elastically coupled, i.e., separated by a spring (or damper), this is treated in DYSCO by including the spring as an element of one of the components or by treating the spring as a separate component.

In the case of a helicopter rotor component, the hub degrees of freedom should be in the fixed system so they may be coupled to the pylon. The blade degrees of freedom should be in the rotating system for ease of interpretation and so that they may be coupled to a rotating control system element. This formulation results in periodic coefficients in a rotor component equation which is usual and quite consistent with equation 1.

As a simple example of coupling, consider three simple components with degrees of freedom as shown in Figure 2. The data for each component follow.

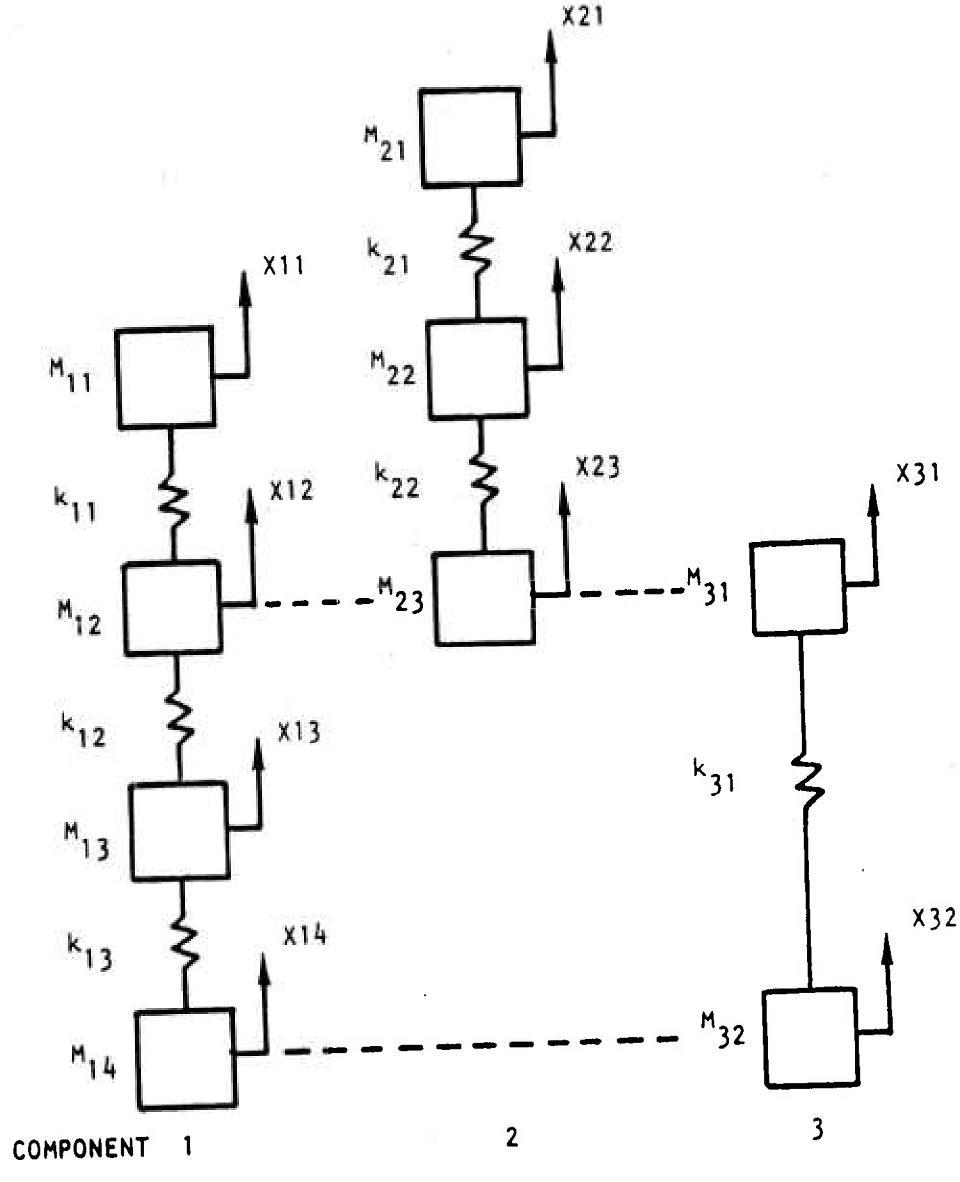


Figure 2. Simple Coupled Structure.

Component 1

$$M_1 = \begin{bmatrix} m_{11} & & & \\ & m_{12} & & \\ & & m_{13} & \\ & & & m_{14} \end{bmatrix}$$

$$K_1 = \begin{bmatrix} k_{11} & -k_{11} & 0 & 0 \\ -k_{11} & k_{11}+k_{12} & -k_{12} & 0 \\ 0 & -k_{12} & k_{12}+k_{13} & -k_{13} \\ 0 & 0 & -k_{13} & k_{13} \end{bmatrix}$$

$$X_1 = \begin{Bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \end{Bmatrix}$$

Component 2

$$M_2 = \begin{bmatrix} m_{21} & & \\ & m_{22} & \\ & & m_{23} \end{bmatrix}$$

$$K_2 = \begin{bmatrix} k_{21} & -k_{21} & 0 \\ -k_{21} & k_{21}+k_{22} & -k_{22} \\ 0 & -k_{22} & k_{23} \end{bmatrix}$$

$$X_2 = \begin{Bmatrix} X_{21} \\ X_{22} \\ X_{23} \end{Bmatrix}$$

<u>Component 3</u>

$$M_3 = \begin{bmatrix} m_{31} & \\ & m_{32} \end{bmatrix}$$

$$K_3 = \begin{bmatrix} k_{31} & -k_{31} \\ -k_{31} & k_{31} \end{bmatrix}$$

$$X_3 = \begin{Bmatrix} X_{31} \\ X_{32} \end{Bmatrix}$$

The components are coupled (joined) at the dashed lines in the figure. The independent degrees of freedom of the "model" may be written

$$X = \begin{Bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{21} \\ x_{22} \end{Bmatrix}$$

The transformation matrices, then, are:

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The model matrix coefficients from Equations 1 - 4 are:

$$M = \begin{bmatrix} m_{11} & & & & & \\ & m_{12}+m_{23}+m_{31} & & & & \\ & & m_{13} & & & \\ & & & m_{14}+m_{32} & & \\ & & & & m_{21} & \\ & & & & & m_{22} \end{bmatrix}$$

$$K = \begin{bmatrix} k_{11} & -k_{11} & 0 & 0 & 0 & 0 \\ -k_{11} & \begin{matrix} k_{11}+k_{12} \\ +k_{31}+k_{22} \end{matrix} & -k_{12} & -k_{31} & 0 & -k_{22} \\ 0 & -k_{12} & k_{12}+k_{13} & -k_{13} & 0 & 0 \\ 0 & -k_{31} & -k_{13} & k_{13}+k_{31} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{21} & -k_{21} \\ 0 & -k_{22} & 0 & 0 & -k_{21} & k_{21}+k_{22} \end{bmatrix}$$

$$F = \begin{Bmatrix} f_{11} \\ f_{12}+f_{23}+f_{31} \\ f_{13} \\ f_{14}+f_{32} \\ f_{21} \\ f_{22} \end{Bmatrix}$$

This illustration represents a very simple case. More complex coupled systems will be described below. Note, however, that all these operations are invisible to the user of the program. When the components and couplings are defined, all operations are automatically performed by the Executive.

1.2.2 Degree of Freedom Names. The procedure used in the program to define the couplings, compute the transformation matrices, and compute the matrices of the coupled system involves the recognition of the names of the degrees of freedom by the Executive.

All degrees of freedom are represented by an 8-character name in FORTRAN format A4,I4. The naming of degrees of freedom is one of the functions of the component technology modules. In some cases, the names are automatically assigned; in other cases, they are provided as user input.

Examples of degree of freedom names are:

BETA12ØØ	Flapping angle, rotor 1, blade 2
PTCH2ØØØ	Pitch angle of structure 2
QFUS13ØØ	Amplitude of mode 3 of structure 1
X12ØØØØØ	User supplied name.

1.2.3 Explicit Coupling. The simplest automatic coupling in DYSCO occurs when the Executive detects two or more degrees of freedom with identical names.

If, in the example on Figure 2, the user had named the degrees of freedom as shown in Figure 3, no further information would be required and all computations would be performed by the program.

It is important that the user be aware of the DOF names automatically assigned to certain components, so that he may conveniently attach other components as desired. The component descriptions in paragraph 3.1 include the definitions of these degree of freedom names.

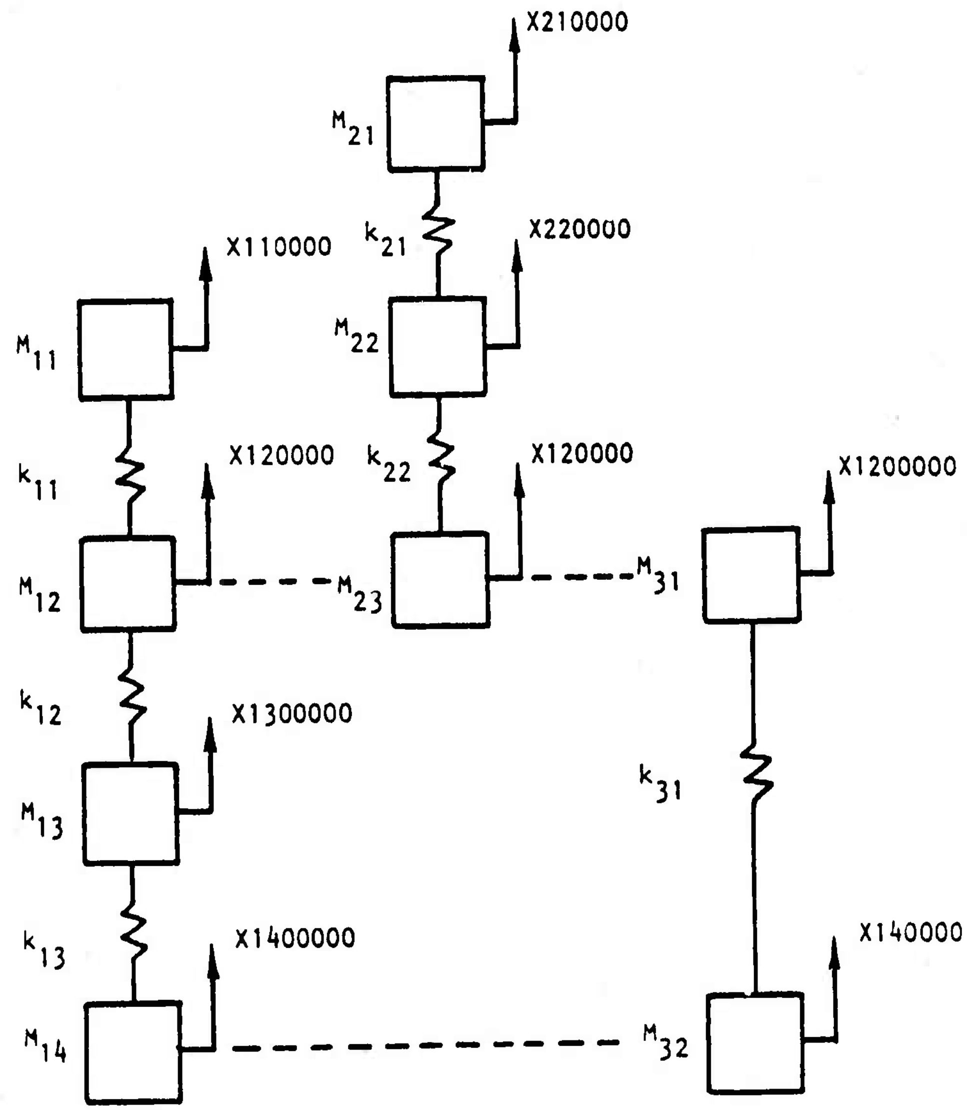


Figure 3. Simple Coupled Structure, DYSCO DOF Names.

As an example of such an application, if the user wished to add a flapping spring to blade 2 of rotor 1, he would provide information (to component CSF1) as follows:

DOF name = BETA12ØØ

M = C = F = 0

K = [k]

When this component is added to the model, the coupling is carried out automatically.

1.2.4 Implicit Coupling. DYSCO also couples structures through linear relationships between degrees of freedom. Certain of the component technology modules will automatically form such relationships based on user input. A particular module (CLC1) allows the user to input arbitrary relationships.

As a very simple example, the user could have coupled the components in Figure 2 by supplying the relationships (using DYSCO naming convention)

X23ØØØØØ = X12ØØØØØ
X31ØØØØØ = X12ØØØØØ
X32ØØØØØ = X14ØØØØØ

instead of renaming as in Figure 3. The results of the two approaches will be identical.

Consider the structure of Figure 3 attached to a modal model as shown in Figure 4. In the figure, ZCG 1ØØØ and PTCH1ØØØ represent the vertical displacement and pitch angle at the center of mass, and QFUS11ØØ, QFUS12ØØ are the modal generalized degrees of freedom of the two modes representing the elastic structure.

The amplitudes of the two mode shapes at a and b are $\phi_1(a)$, $\phi_2(a)$, $\phi_1(b)$, $\phi_2(b)$. Then, the linear relationships that couple these structures are:

$$\text{X11ØØØØØ} = \text{ZCG 1ØØØ} + a * \text{PTCH1ØØØ} + \phi_1(a) * \text{QFUS11ØØ} + \phi_2(a) * \text{QFUS12ØØ} \qquad (5)$$

$$\text{X22ØØØØØ} = \text{ZCG 1ØØØ} + b * \text{PTCH1ØØØ} + \phi_1(b) * \text{QFUS11ØØ} + \phi_2(b) * \text{QFUS12ØØ} \qquad (6)$$

These two relationships are all the information required for the Executive to couple the elastic component to the rest of the system. The user may separately specify these relationships, or in this particular case, they may be formed automatically (see description of CFM2 in Section 3).

During the formation of the transformation matrices, the degrees of freedom on the left side of such relationships are eliminated from the system degrees of freedom. These quantities are called "implicit degrees of freedom" in DYSCO. In such cases, the transformation matrices contain values other than 1 and 0, as in the simple case previously shown.

When the user supplies relationships such as Equations 5 and 6, he may select which degrees of freedom to eliminate from the coupled system by rewriting the equations to place those of his choice on the left side. This may require the hand solution of a small set of algebraic equations. If he wished to eliminate the modal degrees of freedom, Equations 5 and 6 could be rewritten as shown below.

$$\text{QFUS1100} = \bar{\phi}_2(b) * \text{X11ØØØØ} - \bar{\phi}_2(a) * \text{X22ØØØØØ} + (- \bar{\phi}_2(b) + \bar{\phi}_2(a)) * \text{ZCG 1ØØØ} + (- a * \bar{\phi}_2(b) + b * \bar{\phi}_2(a)) * \text{PTCH1ØØØ}$$

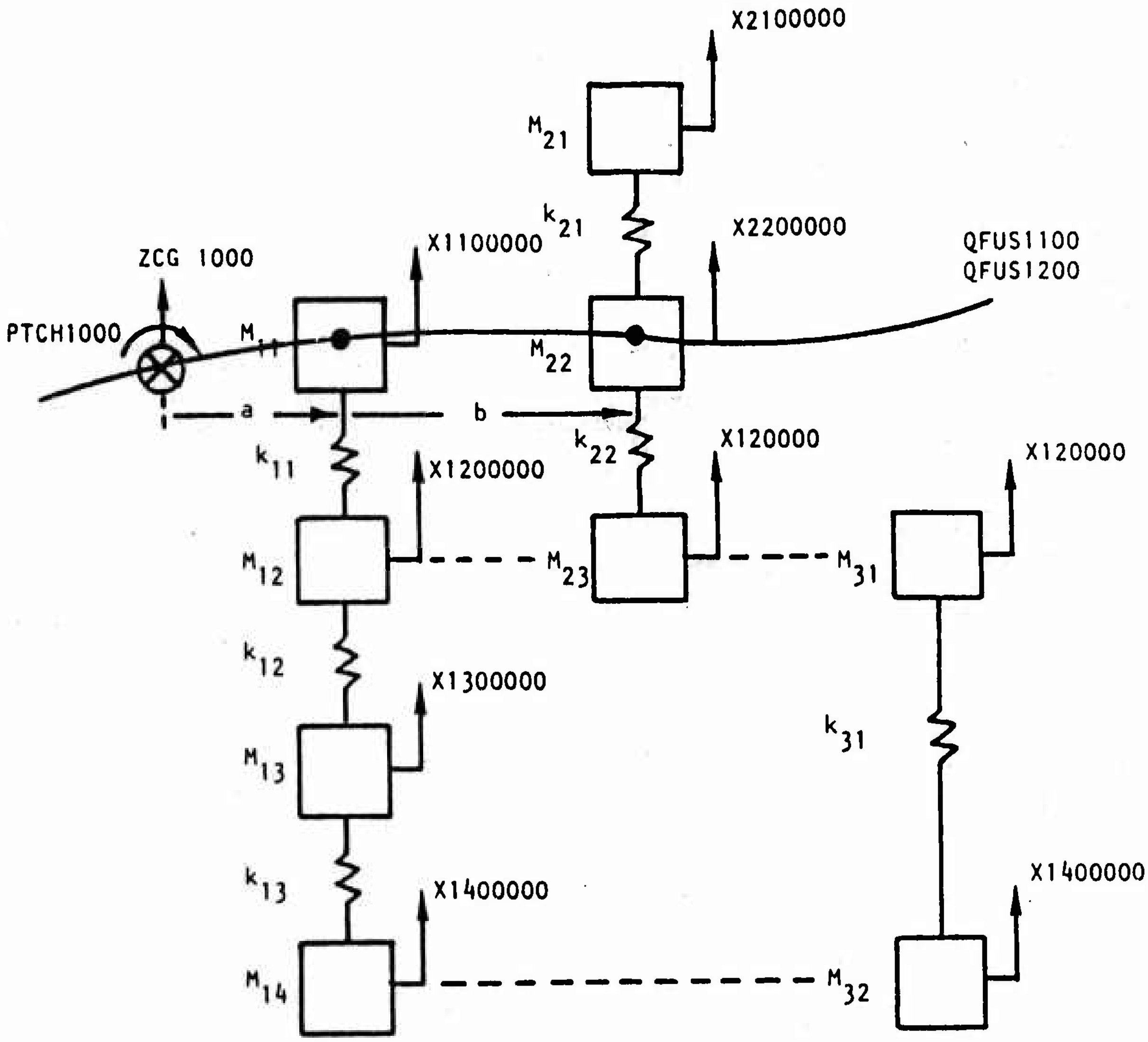


Figure 4. Modal Structure Added to Structure of Figure 3.

$$\text{QFUS1200} = -\bar{\phi}_1(b) * \text{X11ØØØØ} + \bar{\phi}_1(a) * \text{X22ØØØØØ}\ (-\bar{\phi}_1(b) + \bar{\phi}_1(a))$$

$$* \text{ZCG 1ØØØ} + (a * \bar{\phi}_1(b) - b * \bar{\phi}_1(a)) * \text{PTCH1ØØØ}$$

where ($\bar{\ }$) indicates division by $\phi_1(a)\ \phi_2(b) - \phi_2(a)\ \phi_1(b)$.

Once the relationships are specified or automatically formed, all operations, including the formation of the transformation matrices, are invisible to the user.

1.2.5 <u>Other Uses of Implicit Relationships</u>. The linear relationships which replace the degree of freedom on the left by the relationship on the right have uses other than what is usually thought of as coupling of structures. Some examples follow:

1. Eliminate a degree of freedom (dof)

   X14ØØØØØ = Ø * (arbitrary dof)

   This eliminates this dof from the equations. In this case, it represents a cantilever constraint.

2. Change units

   X14ØØØØ = 12 * X14FØØØØ

   Output will be in feet, rather than inches

   PTCH1ØØØ = 1/L * XLØØØØØØ - 1/L * ZCG ØØØØ

   PTCHØØØØ will be replaced by vertical displacement at station L.

3. Represent a mechanical linkage

   BETA1100 = - 1 * BETA1200

   Conversion of an articulated rotor into a teetering rotor.

Note that the degrees of freedom on the right do not have to be degrees of freedom of any component. However, only one new dof may be created for each one that is eliminated.

1.3 MODEL

The physical system which is analyzed in DYSCO (i.e., by solution modules) is termed a "model." The model is defined by specifying component (C---) and force (F---) technology modules and the data set (ds) to be associated with each. The model itself is also stored in the modeling data library where the ds is user-supplied and dm is MODEL.

1.3.1 Component, Force Definition. Each line of a model definition contains the following information:

1. Component technology module name (C---)
2. When necessary, a "structure" or "rotor" number
3. ds of data for C---
4. Force technology module name (F---) or "NONE"
5. ds of data for F---.

When degree of freedom names are automatically assigned, they are tagged with a structure or rotor number to distinguish them from degrees of freedom formed by multiple uses of a component in a model. Structure and rotor numbers must each be unique in any model, except in the case of a rotor control system which must have the same number as the rotor it controls and in the case of a damaged rotor which also must have the same number as the rotor with which it is associated. During input, only necessary data is prompted and only valid input is allowed.

1.3.2 Model Definition. A model definition consists of one or more component, force definitions as in the previous paragraph. Each component and force

may be used more than once with the same or different ds. In general, the order is immaterial, except in special cases such as a rotor control system, which must follow the rotor it is controlling.

The model for the system shown on Figure 4 could be as follows, including a sinusoidal shaker acting on X21ØØØØØ:

INDEX	COMP	NO.	DATA SET	FORCE	DATA SET
1	CSF1		COMP1	NONE	
2	CSF1		COMP2	FSS1	FX21
3	CFM2	1	BEAM	NONE	
4	CSF1		COMP3	NONE	

1.3.3 Model Formation. DYSCO is a command-driven, interactive program (see Section 2 for specific details). The command NEW allows the user to create data sets for components and forces and to create a model definition, as above.

During the creation of a model, the Executive assures that the specified data sets exist and that the force modules are appropriate for the specified component; it determines if any auxiliary data sets are required (such as airfoil tables) and if any "global data" is used (such as wind velocity).

During the execution of the command RUN, the Executive allows for temporarily changed parameters in the data sets. The program then establishes the system degrees of freedom and the transformation matrices, computes the component M, C, K, F matrices, transforms them to the coupled system equations, and allows the user to select a solution algorithm (S---) to be applied to the equations of the model.

1.3.4 Model Details. Prior to the execution of the solution, the user may obtain certain details of the model which may be of interest, as follows:

1. For each component, a list of all the component degrees of freedom
2. A list of all the system degrees of freedom
3. A list of "implicit coefficients" (see directly below)
4. An optional listing of each of the system M, C, K, F constant matrices.

Associated with each component degree of freedom (1, above) is an integer, n:

If $n > 0$, the component dof is system dof number n
If $n = 0$, the dof has been eliminated from the model.
If $n < 0$, the dof has been replaced by a linear relationship to be found in the "packed" table of implicit coefficients.

As an example of the case of $n < 0$, component 2 in paragraph 1.3.2 would include the representation of Equation 6:

2	X21ØØØØØ	X22ØØØØØ	X12ØØØØØ
	(4)	(-5)	(1)

IMPLICIT COEFFICIENTS

	COEF	DOF
5	1.0	ZCG 1ØØØ
6	b	PTCH1ØØØ
7	ϕ_1(b)	QFUS11ØØ
8	ϕ_2 (b)	*QFUS12ØØ

The series starts with element 5 and ends as indicated by the * on the DOF. The next relationship would start with element 9. In an actual case, b, ϕ_1(b), ϕ_2(b) would be numerical values.

1.4 USE OF DYSCO

In this general overview, some of the basic concepts regarding the theoretical basis and the implementation were presented. Following sections include a description of how the program is operated, details regarding all the presently implemented technology modules, and the procedure for adding new technology modules to the library.

2.0 OPERATION OF PROGRAM

2.1 RUN INITIATION

2.1.1 Beginning Execution. DYSCO 4.1 is installed on two mainframes: the VAX under the VMS operating system and the IBM 4341 under the CMS operating system. For both systems, all file assignments are made after DYSCO is initiated; however, each has a different internal procedure for performing the assignments and starting DYSCO execution. Although the procedures differ, user dialogue with DYSCO is very similar.

On the VAX, DYSCO is brought into execution the same as any other program using the VMS command RUN and the program name specified by the installation. All file assignments are performed after execution begins under control of the DYSCO program which prompts the user for required information.

On the IBM, a special "EXEC" file is executed first to prompt the user for file information via JCL commands and second to place the DYSCO program into execution. The user initiates a DYSCO run simply by specifying the name of this EXEC file at the CMS level. The JCL stream initiates dialogue with the user for file assignments and automatically places DYSCO into execution.

2.1.2 File Assignments. There are six types of files which may be assigned to a DYSCO run: the Run Data File; User Data File(s); Sequential File(s); Plot File(s); Load File; and Utility File.

2.1.2.1 Run Data File (RDF) - The RDF is a temporary random direct access file available to the Executive or the user during a single DYSCO run. It is automatically assigned to a specific logical file unit at the beginning of the run and is initialized to null. It is used by the Executive for temporary storage as needed; these usages are not visible to the user. The user may use the RDF to store such data items as Model, Case, Airfoil table, and input data for a Component or a Force Technology Module which are built and used during

the current run only and are not to be saved for a later run. This file is referenced by the name R. Except for the null initialization and temporary nature, the RDF is identical to the User Data File in structure, usage, and internal manipulations by Executive utilities.

2.1.2.2 User Data File (UDF) - The UDF is a random direct access file used to store data which is built and used during a DYSCO run and saved for a later run. Data items such as Model, Case, Airfoil table, and input data for a Component or a Force are stored on a UDF. The number of UDFs which may be defined during a run is zero up to a maximum number which is an installation parameter. At the beginning of the run, the user is prompted for the number of UDFs, followed by a prompt for the file names. The user supplies only the first part of the full file name; DYSCO automatically supplies an installation dependent suffix. These files are internally assigned to sequential logical units. For each UDF, the user is given the option to initialize the file as null (i.e., create a new UDF), or to use and add to a file previously created during a DYSCO run. These files are referenced by the names U1, U2, and so on. Usage of a UDF is strictly under control of the user; the Executive never uses this file for other purposes. Except for the optional initialization and permanent nature, a UDF is identical to the RDF in structure, usage, and internal manipulations by Executive utilities.

2.1.2.3 Sequential File (SF) - A Sequential File is one which is not created or maintained by the Executive. The format of this file is not standardized and is dependent on the specific usage. It may have been created externally to DYSCO, such as an airfoil or induced velocity table, or it may be created internal to the DYSCO environment, but under direct control of specially developed code in the Technology Library. The user is prompted for the number of Sequential Files, which may be zero up to an installation parameter, followed by a prompt for the full file name. These files are assigned sequentially to logical file units. The user references these files as S1, S2, and so on.

2.1.2.4 Plot File (PF) - A Plot File is a permanent sequential file used to store data generated by certain DYSCO Solution Modules in a standardized format, but is not created or maintained by the Executive. It is created internally and is under the direct control of the Solution Module. The number of Plot Files which may be defined during a run is zero up to a maximum number which is an installation parameter. The user is prompted for the number and names of the Plot Files, but supplies only the first part of the full file name; DYSCO automatically supplies an installation-dependent suffix. These files are assigned sequentially to logical file units and are referenced as IPLOT1, IPLOT2, etc. The user can add data from different solutions to a file, but cannot add to a file created during a previous DYSCO run.

2.1.2.5 Loads File (LF) - The Loads File is a temporary sequential file used to store time history data prior to further processing for time history loads calculations. It is created internally and is under the direct control of time history Solution Modules. Only one Loads File can be defined during a run and does not have a name specified by the user. It is assigned to a specific logical file unit and is referenced by the name ILOAD. The file is rewound at the beginning of a time history solution; therefore, only one set of data may be stored on it. The file is erased at the end of the DYSCO run.

2.1.2.6 Utility File (UF) - The Utility File is a temporary file available to the Executive during a DYSCO run which can also be used for temporary data storage by Solution Modules. It is automatically assigned to a specific logical file unit at the beginning of a DYSCO run and is referenced by the name IUTIL. Like the Run Data File and the Loads File, the Utility File is erased at the end of the DYSCO run.

2.1.2.7 Installation Parameters -

FILE	DEVICE NO.	NAME IBM	NAME VAX
U1	1	name D41	name.D41
U2	2	name D41	name.D41
U3	3	name D41	name.D41
U4	4	name D41	name.D41
IN	5	(input)	
OUT	6	(output)	
	7	(punch)	
S1	8	name filetype	name.filetype
S2	9	name filetype	name.filetype
IPLOT1	10	name PLT1	name.PLT1
IPLOT2	11	name PLT2	name.PLT2
ILOAD	12	TEMPX LOAD	TEMPX.LOAD
R	13	TEMPX D41	TEMPX.D41
IUTIL	14	TEMPXX D41	TEMPXX.D41

2.2 COMMANDS

The DYSCO Executive Control System prompts the user for information and, thus, the primary mode of dialogue with the user is of the conversational "question and answer" form. At the highest level of dialogue, however, DYSCO utilizes a command mode where, in response to the query "COMMAND," the user responds with one of the following: CASE, COPY, CRE, DEL, EDIT, LIST, LOOK, NEW, RERUN, RUN, TOC, VAL, and QUIT. A brief description of each of these commands is given in Table 1. These high level commands indicate to DYSCO the nature of the task to be performed, with lower level prompts being issued as needed for additional instructions. Of these commands, those which supply the basic modeling capabilities are NEW and RUN. The remaining commands are auxiliary in nature and offer support to the basic capabilities.

TABLE 1. DESCRIPTION OF COMMANDS

COMMAND	DESCRIPTION
CASE	Forms Batch File for solution to model
COPY	Copies DS/DM from one file to another
CRE	Creates DS/AIRFOIL for airfoil table
DEL	Deletes DS/DM from a file
EDIT	Allows user to modify model Allows user to modify Component/Force input data set
LIST	Prints data summary for any DS/DM
LOOK	Debugging aid for files
NEW	Allows user to create new model Allows user to create new Component/Force input data
RERUN	New solution for model just run
RUN	Forms equations of model and executes a solution
TOC	Table of contents of file (DS/DM description)
VAL	Validates input tables for Components and Forces
QUIT	Terminates DYSCO

2.2.1 CASE. The CASE command allows execution of a run environment created and saved during a previous RUN or RERUN command. The user is prompted for the Case name (the ds name). When ds/CASE is located on the data base, it is read and the core environment is restored to readiness for solution execution. All the information required for the model, coupled system, and solution input values are contained on ds/CASE. No auxiliary or sequential files are required for execution of the CASE.

The intent of the Case capability is to allow the user means to interactively enter all input required to execute a solution for a model, but without the actual execution, which could require a good deal of time. Having provided all input, the user may then execute the solution via a batch-type run.

2.2.2 COPY - The COPY command allows an existing ds/dm to be copied to a different file (RDF/UDF) or to the same file. The user is prompted for the name of the ds/dm to be copied. The Executive locates the ds/dm on the data base and prompts the user for the destination file and the new ds name (the new dm name is the same as the old dm name). After validating the uniqueness of the new ds/dm name with respect to the destination file, the copy is performed.

2.2.3 CRE - The CRE command creates an airfoil table in DYSCO format as a ds/AIRFOIL on an RDF/UDF. The input is a sequential file containing the Airfoil Table values in format consistent with requirements given below. The user is prompted for the sequential file (e.g., S1), the ds name, and the destination file (e.g., R or U1). The ds name must be unique with respect to other airfoil tables on the destination file.

The FORTRAN statements used to write an airfoil table on a sequential file in proper format prior to the DYSCO run are as follows:

```
REWIND  NFILE
WRITE (NFILE,100) NOALFA, NOMACH, (IHEAD(I), I=1,18)
```

```
      WRITE (NFILE,200) ((AMACH(J), ALPHA)(I),CL(I,J),CD(I,J),
                          CM(I,J),
                          I=1,NOALFA), J=1,NOMACH)
  100 FORMAT (2I10/18A4,8X)
  200 FORMAT (7X,F6.4,F7.2,3F10.6)
```

The variables are defined as:

NFILE	- Logical unit number of the sequential file
NOALFA	- Number of angles of attack (between 0 and 65)
NOMACH	- Number of Mach numbers (between 0 and 9)
IHEAD	- 72 character user supplied description
AMACH(J)	- Jth Mach number
ALPHA(I)	- Ith angle of attack, deg
CL(I,J)	- Lift coefficient corresponding to the Ith angle of attack and the Jth Mach number
CD(I,J)	- Drag coefficient corresponding to the Ith angle of attack and the Jth Mach number
CM(I,J)	- Moment coefficient corresponding to the Ith angle of attack and the Jth Mach number

NOTE: The angles of attack for each Mach number are from 0 to 360. The chordwise reference is the 1/4 chord.

2.2.4 DEL. The DEL command allows a ds/dm to be deleted from an RDF/UDF. The user is prompted for the ds and dm names and, when the ds/dm is located on a file, the user is given the option to delete or ignore. This provides protection against accidental deletions. If the ds/dm is found on multiple files, the option is given for each file. If a deletion is performed on a file, the user is given the option to compress the file. If the user intends to perform more than one deletion on a given file, the compress option should be taken only following the last deletion; this is for efficiency.

2.2.5 EDIT. The EDIT command allows two options: (1) modify a previously formed model (ds/MODEL), or (2) modify a previously formed component or force input data set (ds/C--- or ds/F---).

2.2.5.1 Editing a Model - By using the EDIT command, an existing model can be edited to form a different model or to make corrections and replace the original model. Once the user specifies the name of the model to be edited, the model is located on the data base and restored to memory for editing. There are three edit phases: (1) the change phase, (2) the addition phase, and (3) the Global Variables update phase.

Change Phase - At the beginning of the change phase, the model is listed with the components sequentially numbered. These sequence numbers will be used in various edit commands. The user has the following options:

R - Replace a component
D - Delete a component
I - Insert a component (insertion is before sequence number specified)
L - List model
N - End change phase.

For the replace and insert, the user must supply the component, the ds name of the input, the number, and the associated force. The new component input (ds/C---) and the force input (ds/F---) are assumed to exist on the data base. However, if they are not found, the user is allowed the option to form the component or force input before editing is continued. After all changes have been completed, the addition phase is entered.

Addition Phase - During this phase, additional components may be appended to the end of the model. For each addition, the user must supply the component, the ds name of the input, the number, and the associated force. If the input for the component or force is not found on the data base, the user will be given the option to pause and form the input data set (ds/C--- or ds/F---). After all additions have been made, the user is given the option to accept the

additions as correct. If not accepted, all the additions are eliminated with the model being restored to its pre-additions status. The process of allowing components to be added to the model is begun anew. This process continues until the user accepts the additions or specifies no additions are to be made. Upon completion of the addition phase, the new model is validated for consistency.

Global Variables Update Phase - After the new model has been validated, any changes to Global Variables may be performed. If there are no Global Variables required for the new model, then this phase will be bypassed. If there are Global Variables required, then there are two possible situations:

1. The set of new Global Variables is identical to those in the original model. The user will be given the option to either keep the original values unchanged or reenter the entire set of values.
2. The set of new Global Variables is different from those required for the original model. The entire set of old values will be erased from memory and the user will be prompted for values for the entire set of new variables.

Upon completion of these three Edit phases, the edited model is considered complete. The user will be prompted for a new data set name for the model and the save file. If the new ds/MODEL and save file are the same as the original, the user will be asked to verify that replacement should take place. For a replacement, the original ds/MODEL is deleted, the file is compressed, and the new ds/MODEL is written to the file. There are no restrictions on the new ds name and new file except that ds/MODEL must not currently exist on the file (except for a replace).

2.2.5.2 Editing a Component/Force Input Data Set - By using the EDIT command, an existing input data set for a component or force (ds/C--- or ds/F---) can be modified to form a different data set or replace the original data set. An

understanding of several concepts of an input data set is important for effective usage of the EDIT capability.

2.2.5.2.1 Edit concepts.

List of Possible Input Variables - Each component/force has associated with it a full list of input variables which may be required for any particular usage. A component/force may give no options to the user and, thus, may require exactly the same group of variables each time it is used. On the other hand, a component/force may give the user several representational options; and, depending on which options the user selects, the group of variables required may change accordingly. Although the specific requirements may change from usage to usage, each variable which could possibly be required is included in the full list. The list is sequential and is always processed in a sequential fashion.

Variable Description - Each variable in the list has descriptors such as type, properties, existence criteria, range constraints, and override range constraint option:

Type - The variable type may be an integer, real, character string, degree of freedom name (i.e., composed of 4-character name and integer), yes/no (i.e., "Y" or "N"), name of an airfoil table, or sequential file description.

Properties - Variables have associated "properties" such as size (i.e., vector or matrix size), whether it is an increasing or decreasing vector, or whether the elements in a vector are unique.

Existence Criteria - Each variable has zero or more conditions which, for a particular usage of the component/force, must be satisfied in order for the variable to "exist." If a variable exists, then the user must supply an input value. If it does not exist, then an input value is not needed and the variable is ignored.

Range Constraints - If a variable does require a value, then the user-supplied value(s) must satisfy zero or more conditions called range constraints. A simple example of a condition is "greater than 0."

Override Option - The range constraints for some variables may be absolute and must not be violated; for others, these constraints may merely be guidelines for a user. Thus, some variables have an "override" option associated with them. If the user supplies an input value which violates the range constraints, then an option will be given to override these constraints and accept the input.

List of Input Values - For any usage of the component/force, some of the variables in the full list of possible variables will "exist" (i.e., existence criteria are satisfied) and, thus, have values while others may not exist and have no value. A list of the input values will appear to the user as a sequentially numbered list of variables which have values. The user should think of this list of existing variables and values as being part of the full list of variables where "exist" bits are turned on or off as appropriate. Thus, the printed list is a result of extracting from the full list only those variables with the "exist" bit turned on.

2.2.5.2.2 Edit passes.

Multiple Passes - Editing a component/force input data set is performed during a "pass" where an old list of input values is changed to produce a new list of input values. The user may perform multiple passes to produce the desired list of input values to be saved. At the beginning of each pass, there is an old list of values and a new list of values (which is empty). The old list is sequentially numbered with the first "existing" variable being 1, the second being 2, and so on. As variables are changed, a new list is built so that at the end of a successful pass there is a complete new list of input values. When the EDIT command is initiated and the first edit pass is begun, the original input values stored on the ds/C--- or ds/F--- become the old list. At

the end of a pass, the user is given the option to continue with another pass (the new list from this pass becomes the old list for the next pass), start the current pass at the beginning (discard the new list), start entirely over with the original input, or terminate the pass process (the user can then save the new list of input).

Sequential Changes - During a pass, the user must change variables in the old list sequentially and in order. For example, variable 2 must be changed before variable 10. A "pointer" will always point to the last variable in the old list which the user changed. Thus, after the user has completed changing one variable and the Editor is waiting on the next user instruction, the pointer is pointing to a variable in the full list of possible variables which has its "exist" bit turned on. As the user requests different variables to be changed, the pointer is used to step through the list from the last variable changed under user request to the next variable requested to be changed. Each variable in the old list is inspected along the way to determine if it should have a value in the new list and, if so, what value it should have. Unless user action is required, the inspections and subsequent entries in the new list are automatically performed and are invisible to the user. After each intervening variable in the list has been inspected and processed, and the pointer reaches the variable which the user requested to be changed, the user is prompted for a new value, along with the message, "change due to user request." The situation could arise, however, that the requested variable no longer satisfies its existence criteria and, thus, cannot have a value in the new list; a message will be printed for the user. The input and range constraint validation procedure is identical to that used when building the original input data set via the NEW command.

Automatic Validations - The integrity of the new input data set being built is insured by automatic validations during the editing process. As the pointer (described above) steps through the intervening variables between the last user-changed variable and the next variable to be changed, the existence criteria and the range constraints are evaluated using the new list of values, built thus far, as needed (e.g., satisfying references in conditions or used

for dimensioning). As each variable is inspected, there are several possibilities:

1. The variable, perhaps, does not have an old value and should not have a new value.
2. The variable did not have an old value, but a new value is required.
3. The variable did have an old value, but a value in the new list is not needed.
4. The variable did have an old value, it does need a value in the new list, and the old value is still acceptable.
5. The variable did have an old value, it needs a value in the new list, but due to other changes, the old value no longer will meet the range constraints or the dimensioning requirements have changed for a vector or matrix.

If a value is not needed in the new list, or if the old value is still acceptable, then appropriate entry is made for the variable in the new list. However, if a new value is needed (i.e., cases 2 and 5), then the user will be prompted for a value with the message "change due to previous edits."

Matrix Changes - A matrix in the old list which exists, and thus, has values, can be edited in either of two ways: (1) input the entire matrix or (2) change specified items. However, the option to choose and the way in which items are specified vary slightly with the type of matrix involved. There are, in general, four types of matrices: null, diagonal, symmetric, and general. If the matrix has an old type of null, then it makes no sense to offer an item edit; for others, the option is given. If the user selects to input the entire matrix, the type of matrix can be changed and the user is prompted for the new values. If the user selects to perform an item edit, the type of matrix cannot be changed and the prompt message will be appropriate for the type of matrix involved. If the matrix is diagonal, the user will be asked only for the row number and new value. For a symmetric matrix, the user will supply the row and column number and the new value; the symmetric element in the matrix will automatically be changed. For a general matrix, the user will supply the row and column number and the new value. Several item edits

can be performed on a matrix. The same matrix item can be changed several times during the same edit sequence; the last change is used. Validations on a matrix are made after all item changes have been completed.

Pass Commands - During a pass, the following commands are available:

O - List old values. The complete list of existing variables sequence numbers, brief descriptions, and values is printed.

E - List new edited values. The new sequence numbers, brief descriptions, and values for existing variables built thus far will be printed.

S - List sequence numbers and variable names for old list.

C - Change a variable name. The user will be prompted for the sequence number of the variable (in old list).

N - None. Consider pass completed. Pointer steps to end of old list.

Q - Quit. The current pass will be immediately terminated in an error condition. The user will have the option to start the pass again, to start with original input, or quit edit completely (without any option to save changes).

2.2.5.2.3 Edit examples. Two examples are given in Figures 5 through 11 which illustrate several of the edit features. The arrow in the examples indicates a user response to a prompt message. Example 1 illustrates dialogue following initiation of the EDIT command, changing a variable value, the automatic prompt for a variable which did not exist in the old input list, and the automatic prompt for a new value which did exist in the old list but is no longer valid. Example 2 illustrates changing a null matrix, an item edit of a symmetric and a diagonal matrix, and use of the Quit command.

2.2.6 LIST. The LIST command allows the contents of any type of ds/dm to be listed in a readable format. These include ds/MODEL, ds/AIRFOIL, ds/C---, ds/F---, or ds/CASE. The user is prompted for the ds and dm names.

```
→ EDIT
  EDIT MODEL (Y OR N)
→ N
  COMPONENT, FORCE, OR N
→ CES1
  DATA SET
→ PAT1
  LIST EDIT COMMANDS (Y OR N)
→ Y
  EDIT BASE OR GLOBAL VARIABLES
  VALUES TO BE EDITED ARE VIEWED AS A SEQUENTIAL LIST
  MULTIPLE PASSES ARE PERFORMED ON THE LIST
  DURING EACH PASS, VARIABLES MUST BE CHANGED SEQUENTIALLY
  EDIT COMMANDS ARE:
       O -  LIST OLD VALUES
       E -  LIST NEW EDITED VALUES
       S -  LIST SEQUENCE NUMBERS AND VARIABLE NAMES
       C -  CHANGE A VARIABLE NAME
       N -  NONE.  PASS COMPLETED
       Q -  QUIT (IGNORE CURRENT PASS)
  NOTE:  FOR EACH PASS, VARIABLE SEQUENCE NUMBERS MAY BE DIFFERENT.
         IF THERE IS ANY QUESTION,
         GET A LIST AT BEGINNING OF EACH PASS
  ----

  CURRENT DESCRIPTION FOLLOWS
  TEST CES1
  CHANGE DESCRIPTION (Y OR N)
→ N

  NEXT EDIT COMMAND

→ O

    1  MCDF      - # OF DOF-EXCEPT BASE=           4
    2  CDFLI     - (DOF) DOF NAMES
                AAAA   1  BBBB   0  CCCC   0  DDDD   1
    3  BASE      - EXISTNCE OF BASE DOF=          NO
    4  C1        - UPPER DAMPING COEFF =  5.00000E-01
    5  C2        - LOWER DAMPING COEFF =  1.00000E-01
    6  K1        - UPPER SPRING COEFF  =  2.00000E+00
    7  K2        - LOWER SPRING COEFF  =  2.00000E+00
    8  DELT1     - UPPER GAP SIZE      =  3.00000E+00
    9  DELT2     - LOWER GAP SIZE      =  4.00000E+00
  *********************************************************************
```

Figure 5. EDIT Example 1 - Setup.

```
  NEXT EDIT COMMAND
→ C
  SEQUENCE NUMBER OF VARIABLE TO BE CHANGED
→ 4
  ----
     4  C1       START CHANGE DUE TO USER REQUEST
  ----
→ C1    (REAL)
       UPPER DAMPING COEFF
  ENTER 1 REAL VALUE
  ?
→ 8
  NEXT EDIT COMMAND
→ C
  SEQUENCE NUMBER OF VARIABLE TO BE CHANGED
→ 8
  ----
     8  DELT1    START CHANGE DUE TO USER REQUEST
  ----
  DELT1    (REAL)
       UPPER GAP SIZE
  ENTER 1 REAL VALUE
  ?
→ 8
  NEXT EDIT COMMAND

→ N
  EDIT PASS  1 SUCCESSFUL.  LIST NEW VALUES (Y OR N)
→ Y

    1  MCDF      - # OF DOF-EXCEPT BASE=           4
    2  CDFLI     - (DOF) DOF NAMES
               AAAA   1  BBBB   0  CCCC   0  DDDD   1
    3  BASE      - EXISTNCE OF BASE DOF=          NO
    4  C1        - UPPER DAMPING COEFF =  8.00000E+00
    5  C2        - LOWER DAMPING COEFF =  1.00000E-01
    6  K1        - UPPER SPRING COEFF  =  2.00000E+00
    7  K2        - LOWER SPRING COEFF  =  2.00000E+00
    8  DELT1     - UPPER GAP SIZE      =  8.00000E+00
    9  DELT2     - LOWER GAP SIZE      =  4.00000E+00
  *******************************************************************
  ----
  SELECT OPTION
          1 - EDIT COMPLETE
          2 - RESTART THIS EDIT PASS
          3 - RESTART AT BEGINNING WITH ORIGINAL VALUES
          4 - CONTINUE WITH NEXT PASS
→ 4
  NEXT EDIT COMMAND
```

Figure 6. EDIT Example 1 - Change Values.

```
→ C
  SEQUENCE NUMBER OF VARIABLE TO BE CHANGED
→ 3
  ----
     3  BASE      START CHANGE DUE TO USER REQUEST
  ----
   BASE            (Y OR N)
       EXISTNCE OF BASE DOF
  ENTER 1 Y OR N VALUE
→ Y
  NEXT EDIT COMMAND
→ N
  ----
  *NEW* CDFLBI    START CHANGE DUE TO PREVIOUS EDITS
  ----
  CDFLBI    (DOF)
       BASE DOF NAME
  ENTER 1 DOF VALUE (A4,I4)
→ AAAA
  EDIT PASS    2 SUCCESSFUL. LIST NEW VALUES (Y OR N)

→ Y

    1  MCDF       - # OF DOF-EXCEPT BASE=              4
    2  CDFLI      - (DOF) DOF NAMES
                AAAA   1  BBBB    0  CCCC    0  DDDD    1
    3  BASE       - EXISTNCE OF BASE DOF=            YES
    4  CDFLBI     - BASE DOF NAME        =      AAAA    1
    5  C1         - UPPER DAMPING COEFF =   8.00000E+00
    6  C2         - LOWER DAMPING COEFF =   1.00000E-01
    7  K1         - UPPER SPRING COEFF  =   2.00000E+00
    8  K2         - LOWER SPRING COEFF  =   2.00000E+00
    9  DELT1      - UPPER GAP SIZE      =   8.00000E+00
   10  DELT2      - LOWER GAP SIZE      =   4.00000E+00
  *********************************************************************
  ----
  SELECT OPTION
          1 - EDIT COMPLETE
          2 - RESTART THIS EDIT PASS
          3 - RESTART AT BEGINNING WITH ORIGINAL VALUES
          4 - CONTINUE WITH NEXT PASS
→ 4
  NEXT EDIT COMMAND
```

Figure 7. EDIT Example 1 - Automatic New Variable.

```
→ C
  SEQUENCE NUMBER OF VARIABLE TO BE CHANGED
→ 1
  ----
     1  MCDF      START CHANGE DUE TO USER REQUEST
  ----
   MCDF      (INTEGER)
       # OF DOF-EXCEPT BASE
  ENTER     1 INTEGER VALUE
→ 2
  NEXT EDIT COMMAND
→ N
  ----
    2   CDFLI     START CHANGE DUE TO PREVIOUS EDITS
  ----
  CDFLI   (DOF)
       DOF NAMES
  ENTER 2 DOF VALUES (A4,I4) ONE PER LINE
→ AAAA  1
→ BBBB  1
  EDIT PASS   3 SUCCESSFUL. LIST NEW VALUES (Y OR N)
→ Y
    1  MCDF       - # OF DOF-EXCEPT BASE=            2
    2  CDFLI      - (DOF) DOF NAMES     =     AAAA    1     BBBB   1
    3  BASE       - EXISTNCE OF BASE DOF=         YES
    4  CDFLBI     - BASE DOF NAME       =     AAAA    1
    5  C1         - UPPER DAMPING COEFF =  8.00000E+00
    6  C2         - LOWER DAMPING COEFF =  1.00000E-01
    7  K1         - UPPER SPRING COEFF  =  2.00000E+00
    8  K2         - LOWER SPRING COEFF  =  2.00000E+00
    9  DELT1      - UPPER GAP SIZE      =  8.00000E+00
   10  DELT2      - LOWER GAP SIZE      =  4.00000E+00
  ***********************************************************************
  ----
  SELECT OPTION
          1 - EDIT COMPLETE
          2 - RESTART THIS EDIT PASS
          3 - RESTART AT BEGINNING WITH ORIGINAL VALUES
          4 - CONTINUE WITH NEXT PASS
→ 1
  EDITED CES1     TO BE SAVED (Y OR N)
→ Y
  DATA SET FOR EDITED CES1
→ PAT1
  SAVE FILE FOR EDITED CES1
→ U1
  VERIFY ORIGINAL CES1      TO BE REPLACED (Y OR N)
→ Y
  EDIT COMPLETE
  COMMAND
```

Figure 8. EDIT Example 1 - Automatic Prompt New Value.

```
→ 0
    1 NCDF        - NUMBER OF DOF      =          3
    2 CDFLI       - (DOF) DOF NAME
               AAAA    1     BBBB    2     CCCC    3
    3 CM          - (REAL) MASS MATRIX VALUES
               NULL MATRIX
    4 CC          - (REAL) DAMPING MATRX VALUES
               SYMMETRIC MATRIX (LOWER TRIANGLE PRINTED)

    ROW         1
              1.00000E+00
    ROW         2
              2.00000E+00   2.00000E+00
    ROW         3
              3.00000E+00   3.00000E+00   3.00000E+00
    5 CK          - (REAL) STIFFNESS MTRX VALUES
               DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)

       1.00000E+00     2.00000E+00    3.00000E+00
    6 CF          - FORCE VECTOR VALUES
                  1.00000E+00     2.00000E+00    3.00000E+00
  ********************************************************************
  NEXT EDIT COMMAND
→ C
  SEQUENCE NUMBER OF VARIABLE TO BE CHANGED
  3
  ----
    3   CM        START CHANGE DUE TO USER REQUEST
  ----
  CM      (REAL)
       MASS MATRIX VALUES
  TYPE MATRIX
  (0=NULL), (1=DIAGONAL), (2=SYMMETRIC), (3=GENERAL)
→ 1
  INPUT      3 DIAGONAL REAL VALUES
  ?
→ 4.5 4.5 4.5
  NEXT EDIT COMMAND
→ E
    1  NCDF       - NUMBER OF DOF        =            3
    2  CDFLI      - (DOF) DOF NAME
               AAAA    1   BBBB    2  CCCC    3
    3  CM         - (REAL) MASS MATRIX VALUES

               DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)

         4.50000E+00    4.50000E+00    4.50000E+00
  ********************************************************************
```

Figure 9. EDIT Example 2 - Null Matrix.

```
  NEXT EDIT COMMAND
→ C
  SEQUENCE NUMBER OF VARIABLE TO BE CHANGED
→ 4
  ----
     4  CC       START CHANGE DUE TO USER REQUEST
  ----
   CC       (REAL)
       DAMPING MATRIX VALUES
  ITEM EDIT (Y OR N)
→ Y
  SYMMETRIC MATRIX
  ROW NUMBER
→ 1
  COLUMN NUMBER
→ 3
  ENTER 1 REAL VALUE(S)
  ?
→ 4.5
  ELEMENT ( 3, 1) CHANGED ALSO
  ANOTHER ITEM (Y OR N)
→ Y
  SYMMETRIC MATRIX
  ROW NUMBER
→ 3
  COLUMN NUMBER
  2
  ENTER 1 REAL VALUE(S)
  ?
→ 4.5
  ELEMENT ( 2, 3) CHANGED ALSO
  ANOTHER ITEM (Y OR N)
→ N
  NEXT EDIT COMMAND
→ E
    1  NCDF      - NUMBER OF DOF       =            3
    2  CDFLI     - (DOF) DOF NAME
               AAAA   1  BBBB   2  CCCC   3
    3  CM        - (REAL) MASS MATRIX VALUES
               DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
         4.50000E+00   4.50000E+00   4.50000E+00
    4  CC        - (REAL) DAMPING MATRX VALUES
               SYMMETRIC MATRIX (LOWER TRIANGLE PRINTED)
       ROW     1
                   1.00000E+00
       ROW     2
                   2.00000E+00   2.00000E+00
       ROW     3
                   4.50000E+00   4.50000E+00   3.00000E+00
*********************************************************************
```

Figure 10. EDIT Example 2 - Symmetric.

```
  NEXT EDIT COMMAND
→ C
  SEQUENCE NUMBER OF VARIABLE TO BE CHANGED
→ 5
  ----
     5  CK       START CHANGE DUE TO USER REQUEST
  ----
   CK       (REAL)
       STFFNESS MATRIX VALUES
  ITEM EDIT (Y OR N)
→ Y
  DIAGONAL MATRIX
  ROW NUMBER
→ 2
  ENTER 1 REAL VALUE(S)
  ?
→ 4.5
  ANOTHER ITEM (Y OR N)
→ N
  NEXT EDIT COMMAND
→ E
     1  NCDF      - NUMBER OF DOF        =            3
     2  CDFLI     - (DOF) DOF NAME
                AAAA   1  BBBB   2  CCCC   3
     3  CM        - (REAL) MASS MATRIX VALUES
                DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
        4.50000E+00   4.50000E+00   4.50000E+00
     4  CC        - (REAL) DAMPING MATRX VALUES
                SYMMETRIC MATRIX (LOWER TRIANGLE PRINTED)
       ROW     1
        1.00000E+00
       ROW     2
        2.00000E+00   2.00000E+00
       ROW     3
        4.50000E+00   4.50000E+00   3.00000E+00
     5  CK        - (REAL STFFNESS MTRX VALUES
                DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
        1.00000E+00   4.50000E+00   3.00000E+00
  **********************************************************************
  NEXT EDIT COMMAND
→ QUIT
  ERROR -INVALID COMMAND QUIT VALID COMMANDS ARE
       O    E    S    Q    N    C
  NEXT EDIT COMMAND
→ Q
  ERROR - IN EDIT PASS FOR PAT1 /CSF1
  ----
  SELECT OPTION
          1 - EDIT COMPLETE
          2 - RESTART THIS EDIT PASS
          3 - RESTART AT BEGINNING WITH ORIGINAL VALUES
```

Figure 11. EDIT Example 2 - Diagonal.

2.2.7 <u>LOOK</u>. The LOOK command provides information about an RDF/UDF and is useful as a debugging aid. The user is prompted for the file (e.g., R or U1). For this file, the following information is given:

Number of ds/dm entries
First record number of the first ds/dm
Next available record on the file

For each ds/dm, the following information is given:

ds/dm name
First record number
Number of records
Option to print all records for the ds/dm
(only 72 characters of each record are printed)

2.2.8 <u>NEW</u>. The NEW command allows two options: (1) the formation of a new model or (2) the formation of new component and/or force input data sets.

2.2.8.1 <u>New Model</u> - The NEW command may be used to form a new model (ds/MODEL) on a specified RDF/UDF. The new model will be primarily a configuration of components and any associated forces. Auxiliary information implied by the components and forces selected will also be a part of the new model.

The user is prompted for each component to be included in the model and the ds name for its input. The order of specification of components is arbitrary (except for the CCE1 and CCEO components, which must be specified after the rotor they control has been specified). The user may then be prompted for a "component number." Some physical components of the model must be identified by an integer (1 - 9). Rotors and other structures may each have a separate set of numbers. Thus, there may be a rotor number 1 and a structure number 1. A special case is the rotor control system (represented by CCE1 or CCEO) which must have a component number identical to that of the rotor it controls. If appropriate, the user is prompted for the force to be applied and the ds name

for its input. The data base is searched for the component input, ds/C---, and for the force input, ds/F---. If multiples are found, the user selects the proper file. If the component input is not found, then the user may supply data to form a new component before proceeding with forming the model. If the force input is not found, then the user may likewise form a new force before proceeding. If the user elects this option, the process is identical to that described below for forming component/force input via the NEW command. For each force specified, ds/F--- will be inspected for any auxiliary ds/AIRFOIL and sequential file requirements; these requirements become part of the model.

The Global Variable requirements for the model are also determined. Each component and force has an associated list of global variables which are required for model execution. For each component and force specified for the model, the Executive inspects its associated list and forms a "pooled" set of all global variables required. Each global variable may have zero or more associated conditions which must be satisfied in order for that variable to actually require a value. These conditions, or existence criteria, are evaluated for each variable, and for those whose conditions are satisfied the user is prompted for a value(s). Each variable may also have range constraints and, if so, the input value must satisfy these constraints.

If invalid data is detected, the user is given the opportunity to reenter the data. The set of global variables and their values become part of the model.

Upon completion of the model, a summary is printed and the model is written to the RDF/UDF. This ds/MODEL may be used for a solution method execution in a subsequent RUN command. It may also be listed via the LIST command.

2.2.8.2 <u>New Component/Force Input Data Set</u> - Input data for one or more components and/or forces may be formed by the NEW command. The result of a component formation is the creation on an RDF/UDF of ds/C--- which contains the input for the component. The result of a force formation is similarly the

creation of ds/F--- which contains the input for the force. The user is prompted for the name of a component or force. After the name is validated, the user is prompted for all other information and input data needed. The process of prompting the user for the required input is described in the following paragraph.

Each component and force has associated with it a sequential list of base variables; this list is a pool of all variables which are "local" in nature to the component or force and which can be required for any particular definition or usage. Each variable has associated with it zero or more conditions which must be satisfied in order for the variable to be required or "exist;" these are called existence criteria. If a variable has any existence criteria specified, the condition(s) generally involve the value given to a variable earlier in the list or the existence of an earlier variable. The list of base variables is processed sequentially, starting with the first and completely processing each one in turn before continuing to the next variable in the list. In processing a variable, if the existence criteria are satisfied, the user will be prompted for a value(s). If not, the variable is ignored and processing continues with the next variable in the list. Each variable also has associated with it zero or more range constraints, which are conditions which must be satisfied in order for the user input to be accepted. All user-supplied data is validated against these constraints with reentry options given when invalid input is encountered. When the list of base variables has been completely processed, the set of variables which satisfied their existence criteria and their values are considered to be the input for the component or force being defined. These variables and their values are stored on ds/C--- or ds/F---. More detailed information about the input process is provided in paragraph 2.2.5.2. The resulting component or force data set can be used in model formation via a subsequent NEW command. It may also be listed via the LIST command.

2.2.9 <u>RERUN</u>. The RERUN command allows a model just RUN to be efficiently run again with a different solution method or with the same solution method, but

different solution input parameters. The information in memory which was "left over" from the just completed RUN command is utilized, thus allowing the model and coupled system setup phases to be bypassed. Since these setups may require a good deal of disk I/O, bypassing improves efficiency.

2.2.10 RUN.

Run Setup - The RUN command allows a solution method to be specified for a model and either executed or saved for later execution. The user is prompted for the name of the model (i.e., the ds name). The model specified (ds/MODEL) is located on the data base. If found on multiple files, the user selects the proper file. It is read into memory and the data base is searched for all ds/dm requirements (component and force inputs, auxiliary airfoil tables). If any are found on multiple files, the user makes the selection. For any sequential file requirement, the user is prompted for the file (e.g., S1). After satisfying the file requirements, a list of the model, including any global variable values, is given.

Temporary Edits - Following this initial setup, the user is given the option of temporarily editing the global variables and also the input for any component or force. Any changes made in this temporary fashion will be active only until completion of the RUN command (and any immediately executed RERUN). The original data stored with the model or as component or force input will not be affected by a temporary edit. Full editing power obtained with the command EDIT is available to the user during a RUN edit, except with regard to auxiliary airfoil tables and sequential files which may be required by a force. As noted above, the airfoil table and sequential file requirements for the model have already been processed prior to RUN editing and additional requirements cannot be handled properly. Thus, any changes to the input data for a force which result in an additional airfoil table or sequential file being required are not allowed. If this occurs, the current edit pass will be abandoned and the user will be given the option to return to an earlier edit pass or terminate. On the other hand, any changes to the input data for a force which result in an airfoil table or sequential file no longer being required are

acceptable (although the Executive has processed these requirements earlier and insured their availability, they will be unused during the RUN). If this situation occurs, the user will be informed and will be warned that it will be impossible for these requirements to be reactivated by a subsequent edit pass. If the user did not intend for this situation to occur, the current edit pass should not be accepted and the user should start the current edit pass over. These edited values are used in the subsequent coupling process.

Coupling and Solution - Once all temporary editing is completed (if any), the Executive initiates the coupling process and the coupled system matrices (constant) are formed. Following this formation process, the option is given to list the details and matrices. A Solution may then be specified and the user is prompted for the input. The user may choose the option of saving the solution environment as a Case (see CASE command) with a snapshot of relevant parts of the current memory state being saved as a ds/CASE on an RDF/UDF. Even if the Case option is chosen, the user is also given the option to execute the Solution immediately.

2.2.11 TOC. The TOC command prints a partial or complete Table of Contents for the RDF, any UDF, or the entire data base. The user is prompted for a file, a ds name, and a dm name. The character * is a "wild card" response for this command and, when specified, is interpreted as "all." The user can use combinations of * with other responses to perform specific searches or complete TOCs of the entire data base. For each ds/dm printed, a description is also given.

2.2.12 VAL. The VAL command validates various input tables for installed Technology Modules and is a critical debugging and installation aid for new technology. The following tables are automatically validated:

Global Variables Table (IGTAB in /XBG/)
Global Constants Table (IGCTAB in /XBG/)

For each installed component:

Base Variables Table (IBTAB in C---I)
Global Variables Selection Table (IGVTAB in C---I)

For each installed force:

Base Variables Table (IBTAB in F---I)
Global Variables Selection Table (IGVTAB in F---I)

The construction of these tables is analyzed with respect to a variety of correctness rules such as syntax, references to variables of correct type, table entry length, and overall table dimension. The validations performed are detailed and quite encompassing. As each table is validated, a message is printed that it is "ok" or that it is in error. In the case of error, descriptive information is provided. Since a change in the Global tables can have far-reaching impacts on various modules, a full sweep of the tables is required to insure system integrity.

2.2.13 QUIT. The QUIT command terminates the DYSCO run.

2.3 OPERATIONAL SCENARIO

Basic Modeling Scenario - DYSCO is a "domain" executive control system designed to support the basic modeling scenario shown in Figure 12. The underlying process of this scenario is the sequence of steps which a design engineer would take to perform a task in the domain of dynamic structural analysis. First, the engineer would define the component(s) and the force(s) representations, followed by defining a model composed of these components and forces. After the model has been defined, the coupled system would be formed and the input parameters for a solution method would be defined. The last step would be to execute the solution. These steps represent the "ideal" scenario where only one pass through the process would accomplish the analysis goals. Of course, this is rarely the case in reality where backtracking, modifications, and iterations are an assumed part of a design task. This sequential representation is useful, however, in understanding the role of the DYSCO

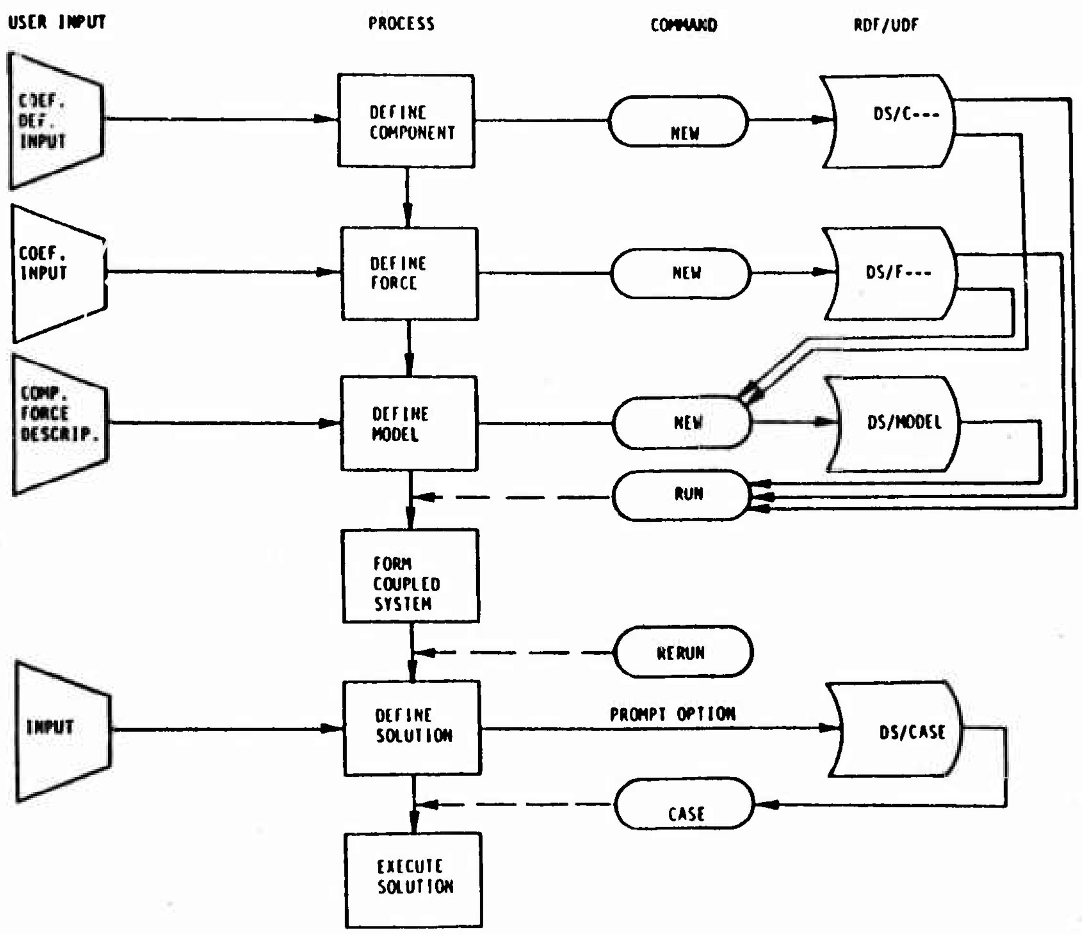


Figure 12. Modeling Scenario and Command Relationship.

modeling commands and how they can be used effectively in iterative design tasks.

Basic Modeling Scenario and Command Relationship - Of the DYSCO commands discussed previously, only four are considered as basic modeling commands: NEW, RUN, RERUN, and CASE. The other commands provide support functions (e.g., editing, copying) for the modeling task. The relationship of the modeling commands to the basic modeling scenario is shown in Figure 12. Of these, only the NEW and RUN commands are essential in performing a design task. RERUN and CASE can be used to improve efficiency.

NEW - NEW provides for defining components, forces, and models. For a component or force, the user is prompted for all input which is validated and stored on a file as ds/C--- or ds/F---. For a model, the user specifies the names of these already-built component/force input data sets to form the model. These may be specified in any order (except restrictions noted in paragraph 2.2.8.2). However, if they have not previously been built, the user will be allowed to pause model formation and build these data sets.

RUN - Once a model has been built, RUN provides for forming the coupled system for this model, defining the solution, and executing the solution.

RERUN - RERUN allows the user to use the same model and execute different solution methods (or the same solution method with different input parameters) in an efficient way. If the model is used to form the coupled system and a solution method is executed, a second solution method can be immediately executed without going through the coupling process a second time. This can be repeated for several different solution methods.

CASE - The CASE capability allows the user to perform everything interactively in the design task, except actually executing the solution, which may be computationally intensive and unsuited for the interactive setting. In a RUN or RERUN command at the point where

the solution method and its input parameters have been defined, all input data stored on files (e.g., component input, airfoil tables) has been read and digested by the system; everything needed for solution execution is in memory. A prompt option is given and the user may elect to snapshot the memory environment and store it on a ds/CASE. The user may later resume execution at this snapshot point via the CASE command in a batch-type environment.

Basic Operational Scenario - The basic operational scenario in which a design task is performed in DYSCO is a three-step procedure using only NEW and RUN:

1. Build Component and Force Input - The NEW command is executed to build the input data sets for each component which is planned to be used in a model. NEW allows the user to build an unlimited number of components and forces without returning to the higher command level. Thus, input data sets for all the forces planned for the model are also built. These can be saved on the same file or separate files at user discretion.

2. Build Model - The NEW command is executed to build the model. The names of the components and forces and their input data sets are specified by the user. The order of specification is not related in any way to the order in which the input data sets were built. The only requirement is that the files which contain the data sets must be attached to the current DYSCO run. Each force input data set is inspected to determine any auxiliary airfoil table requirement or sequential file requirements. These requirements are printed for the user at the end of the model formation. The user may save the new model (ds/MODEL) on any RDF/UDF.

3. Execute Solution - After the model is built, the RUN command is executed. The user must ensure that all component and force input data sets and all airfoil table (ds/AIRFOIL) and sequential file requirements are attached to the current DYSCO run.

The coupled system is formed and the user specifies the solution method to be executed. The user is prompted for all solution input data and the solution is executed. At various points during the RUN, the user is given various options for temporary editing of component/force input data sets, for printing coupled system details and matrices, and for executing or saving the solution environment.

3.0 INSTALLED TECHNOLOGY MODULES

This section contains complete descriptions of all the technology modules available for use. Each has a name of four characters, as follows:

C--- - Component
F--- - Force
S--- - Solution

For each type of technology module, the 2nd and 3rd characters are descriptive and the 4th character is an arbitrary integer which implies a level of complexity (9 being the most complex).

3.1 COMPONENT TECHNOLOGY MODULES

A component module represents an algorithm for computing mass, damping, stiffness matrices, and a force vector for a given formulation of a set of second order differential equations. In addition, degrees of freedom and linear relationships between degrees of freedom are defined.

Except as specifically noted in this description, the following characteristics apply to all component technology modules:

All degrees of freedom are optional.
Any combination of components is valid in a model.
Each component may be used more than once in a model.
The order of components is arbitrary.
Structure or rotor numbers are assigned in model definition for components which require them.
Structure and rotor number must each form a unique set in any model.

During interactive input, the FORTRAN name of the parameter and a descriptive message are displayed. Only required data, based on previous input, is requested. Certain parameters have range constraints. If these are not

satisfied, a message will be displayed requesting a reentry. In certain cases, the user will have an option to override the formal constraint.

3.1.1 CFM2 - Fuselage, Modal. CFM2 is a modal representation of a thin elastic structure. A three-dimensional fuselage-like structure may be modeled relative to coincident principal axes and elastic axes of the structure.

3.1.1.1 Primary Features -

Rigid Body Modes - Up to three translational and three rotational degrees of freedom defined in a Cartesian coordinate system whose origin is the center of gravity of the component.

Elastic Modes - Up to six normal modes with vertical and lateral deflections and slopes and torsional deflections.

Implicit Degrees of Freedom - Up to 12 other displacement degrees of freedom may be defined along the component centerline and may be oriented at angles defined in the X - Z and Y - Z planes (one of the two angles being small). These implicit degrees of freedom are arbitrary and may be degrees of freedom of other components, which will result in the automatic coupling of the components. Implicit and rotor hub degrees of freedom are automatically formulated into linear combinations of the rigid body and elastic degrees of freedom.

Rotor Interfaces - Up to 4 rotors may be automatically coupled through rigid massless connectors (shafts). The rotor hub degrees of freedom are defined as above, but may also include a displacement from the component centerline.

Aerodynamic Forcing - Optional use of FFAØ for flat plate drag or FFC2 for fuselage surface aerodynamic forces (see paragraph 3.2 for details).

Multiple Uses - CFM2 can be used up to 4 times in a given model.

3.1.1.2 Degrees of Freedom - The names of the degrees of freedom chosen by the user are formulated automatically as follows:

Rigid Body Degrees of Freedom -

XCG sØØØ - longitudinal translation
YCG sØØØ - lateral translation
ZCG sØØØ - vertical translation
ROLLsØØØ - roll
PTCHsØØØ - pitch
YAW sØØØ - yaw

Elastic Degrees of Freedom -

QFUSsmØØ

where s = structure number (assigned during model formulation)
m = elastic mode number

The positive orientations of the rigid body and interface degrees of freedom and the literal portions of the names of the rigid body degrees of freedom are presented in Figure 13.

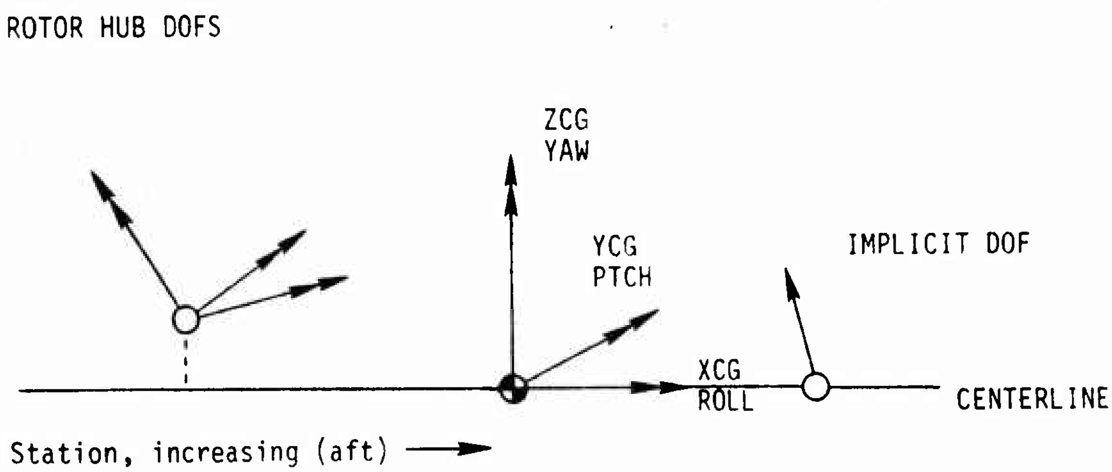


Figure 13. CFM2 Rigid Body and Interface Degrees of Freedom.

3.1.1.3 <u>Input, CFM2</u> - The CFM2 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS (FORTRAN READ FORMAT)
response

Only required input (based on previous input) is requested.

RBM (Y or N)
RIGID BODY MODES
ENTER 1 Y or N VALUE

(Enter Y [yes] if rigid body modes are to be included, else N [no])

IXCG (Y or N)
LONGITUDINAL
ENTER 1 Y or N VALUE

(Y if longitudinal degree of freedom is elected, else N)

IYCG (Y or N)
LATERAL
ENTER 1 Y or N VALUE

IZCG (Y or N)
VERTICAL
ENTER 1 Y or N VALUE

IROLL (Y or N)
ROLL
ENTER 1 Y or N VALUE

IPTCH (Y or N)
PITCH
ENTER 1 Y or N VALUE

IYAW (Y or N)
YAW
ENTER 1 Y or N VALUE

CG (REAL)
CG STATION (IN)
ENTER 1 REAL VALUE

(Station [inches] for CG; only necessary for pitch or yaw degrees of freedom)

NMODE (INTEGER)
NO. OF ELASTIC MODES
ENTER 1 INTEGER VALUE(S)

(Enter the number of elastic modes; $\emptyset \leq$ NMODE ≤ 6)

NS (INTEGER)
NO. FUSELAGE STAS
ENTER 1 INTEGER VALUE(S)

(Enter the number of fuselage stations at which elastic modes will be defined; $5 \leq$ NS ≤ 20)

X (REAL)
INPUT STATION VALUES
ENTER NS REAL VALUES

(Enter the stations [inches] in ascending order)

VCn (Y or N)
MODEn VERTICAL COMP
ENTER 1 Y OR N VALUE

(Enter Y if mode n has a vertical component, else N)

Zn (REAL)
MODEn VERTICAL DISP
ENTER NS REAL VALUE(S)

(Enter the vertical displacement for each station for mode n)

ZPn (REAL)
MODEn VERTICAL SLOPE
ENTER NS REAL VALUE(S)

(Enter the vertical slopes for each station for mode n)

LCn (Y or N)
MODEn LATERAL COMP
ENTER 1 Y OR N VALUE

Yn (REAL)
MODEn LATERAL DISP
ENTER NS REAL VALUE(S)

YPn (REAL)
MODEn LATERAL SLOPE
ENTER NS REAL VALUE(S)

TCn (Y OR N)
MODEn TORSION COMP
ENTER 1 Y OR N VALUE

T (REAL)
MODEn TORSION DISP
ENTER NS REAL VALUE(S)

NR (INTEGER)
NO. OF ROTORS
ENTER 1 INTEGER VALUE(S)

(Enter the number of rotors to be coupled; $\emptyset \leq NR \leq 4$)

NROT (INTEGER)
ROTOR NUMBERS
ENTER NR INTEGER VALUES

(Enter rotor identification numbers; $\emptyset \leq NROT(I) \leq 4$)

XROT (REAL)
ROTOR STATIONS
ENTER NR REAL VALUES

(Enter the stations [inches] at which the rotors are coupled [need not be previously defined fuselage stations])

ZROT (REAL)
ROTOR VERTICAL HT
ENTER NR REAL VALUES

(Enter the vertical heights [inches] of the rotor hubs above the X - Y plane)

ASF (REAL)
FWD SHAFT ANGLE
ENTER NR REAL VALUES

(Enter the angles [degrees] with respect to the component vertical axis in the X - Z plane, positive being defined as inclination toward the negative X-axis, at which the shafts to the rotor hubs are inclined; ASF(I) must be small if ASL(I) is large)

ASL (REAL)
LAT SHAFT ANGLE
ENTER NR REAL VALUES

(Enter the angles [degrees] with respect to the component vertical axis in the Y - Z plane, positive being defined as inclination toward the positive Y-axis, at which the shafts to the rotor hubs are inclined; ASL(I) must be small if ASF(I) is large)

IX (Y OR N)
HUB TRAN DOF - LONG
ENTER NR Y OR N VALUES (35A2)

(Enter Y if the rotor hub longitudinal degree of freedom is included in the rotor component, else N)

IY (Y OR N)
HUB TRAN DOF - LAT
ENTER NR Y OR N VALUES (35A2)

IZ (Y or N)
HUB TRAN DOF - AXIAL
ENTER NR Y OR N VALUES

IAX (Y OR N)
HUB ANGL DOF - ROLL
ENTER NR Y OR N VALUES (35A2)

IAY (Y OR N)
HUB ANGL DOF - PITCH
ENTER NR Y OR N VALUES (35A2)

IAZ (Y OR N)
HUB ANGL DOF - YAW
ENTER NR Y OR N VALUES (35A2)

NI (INTEGER)
NO. OTHER IMPLCT DOF
ENTER 1 INTEGER VALUE(S)

(Enter the number of other implicit displacement degrees of freedom; $\emptyset \leq NI \leq 12$)

CIDFL (DOF)
IMPLICIT DOF NAMES
ENTER NI DOF NAMES (A4,I4) ONE PER LINE

(Enter the implicit degree of freedom names)

XSTA (REAL)
STA FOR EA IMPL DOF
ENTER NI REAL VALUES

(Enter the stations [inches] at which the implicit degrees of freedom are coupled [need not be previously defined fuselage stations])

AF (REAL)
FWD ANGLE FROM VERT
ENTER NI REAL VALUES

(Enter the angles [degrees] with respect to the component vertical axis in the X - Z plane, positive being defined as inclination toward the negative X-axis, at which the implicit degrees of freedom are inclined; AF(I) must be small if AL(I) is large)

AL (REAL)
LAT ANGLE FROM VERT
ENTER NI REAL VALUES

(Enter the angles [degrees] with respect to the component vertical axis in the Y - Z plane, positive being defined as inclination toward the positive Y-axis, at which the implicit degrees of freedom are inclined; AL(I) must be small if AF(I) is large)

MASSL (REAL)
FUSELAGE WEIGHT (LB)
ENTER 1 REAL VALUE

(Enter the fuselage weight [lb])

IMXF (REAL)
ROLL MOI (SLUG-FT(SQ))
ENTER 1 REAL VALUE

(Enter the roll mass moment of inertia of the fuselage)

IMYF (REAL)
PITCH MOI ABOUT CG
ENTER 1 REAL VALUE

IMZF (REAL)
YAW MOI ABOUT CG
ENTER 1 REAL VALUE

MMS (REAL)
MODAL MASS (SLUGS)
ENTER NMODE REAL VALUES

(Enter the modal mass for each elastic mode)

MD (REAL)
MODAL DAMPING (PCT)
ENTER NMODE REAL VALUES

(Enter percent of critical damping for each elastic mode)

FREQ (REAL)
MODAL FREQUENCY (HZ)
ENTER NMODE REAL VALUES

3.1.2 CRR2 - Rotor, Rigid Blades. CRR2 is a rotor system representation with rigid hinged blade degrees of freedom defined in a rotating coordinate system coupled to hub translation and rotation degrees of freedom defined in a nonrotating coordinate system.

3.1.2.1 Primary Features -

Rigid Hinged Blades - Up to 9 identical rigid hinged blades which may have flap, lag, and pitch degrees of freedom. Blade degrees of freedom are defined in a coordinate system which rotates about the hub vertical principal axis.

Coincident Hinges With Springs and Dampers - Flap and lag hinges are located coincidently and may be displaced spanwise from the vertical principal axis. A spring and a damper may be specified for each blade degree of freedom.

Flap-Lag Coupling - A coupling factor may be applied for flap-lag damping coupling and for flap-lag spring coupling.

Control System Interface - The blade pitch degree of freedom may be coupled to a control system component through a rigid pitch horn displaced spanwise from the hinge location.

Uniform or Nonuniform Blades - Option allowed for convenience of input.

Hub Degrees of Freedom - Up to three translational and three rotational hub degrees of freedom defined in a nonrotating Cartesian coordinate system.

Aerodynamic Forcing - Optional use of FRAØ, FRA2, or FRA3 rotor aerodynamic forces (see paragraph 3.2 for details).

Multiple Uses - CRR2 can be used up to 4 times in a given model.

3.1.2.2 Degrees of Freedom - The names of the degrees of freedom chosen by the user are formulated automatically as follows. All are optional, except that at least one blade degree of freedom is required (see Figure 14).

Blade Degrees of Freedom (Rotating System) - one set for each blade:

BETArbØØ - flap angle
ZETArbØØ - lag angle
THETrbØØ - pitch angle

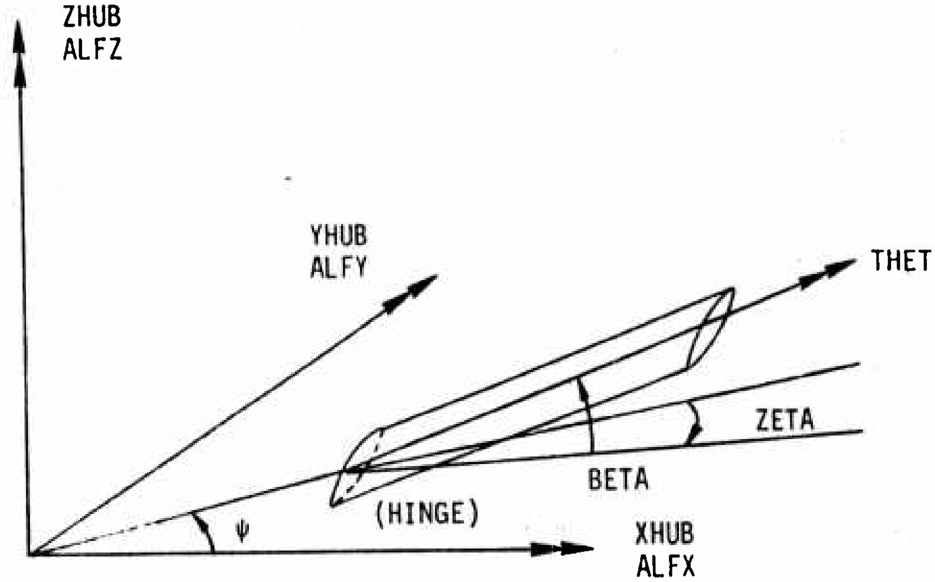


Figure 14. CRR2 Degrees of Freedom.

Hub Degrees of Freedom (Nonrotating System) -

XHUBrØØØ - longitudinal translation

YHUBrØØØ - lateral translation

ZHUBrØØØ - vertical translation

ALFXrØØØ - roll

ALFYrØØØ - pitch

ALFZrØØØ - yaw

where r = rotor number (assigned during model formulation)

b = blade number (blade 1 is the reference blade)

Implicit Degrees of Freedom (Rotating System) - The relationship between control rod degrees of freedom (RODRrbØØ) and blade degrees of freedom is:

RODRrbØØ = PHL * THETrbØØ + PHSTA * BETArbØØ

where PHL is the pitch horn length and PHSTA is the pitch horn station.

The implicit relationship is automatically established if blade pitch is a degree of freedom and PHL ≠ Ø. This is used to automatically couple the rotor component to an associated control system component (CCEØ or CCE1).

The positive orientations of the component degrees of freedom are presented in Figure 14. Only the literal portions of the degree of freedom names are shown. ψ is the azimuth angle of the reference blade (blade 1).

3.1.2.3 Input, CRR2 - The CRR2 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS (FORTRAN READ FORMAT)
response

Only required input (based on previous input) is requested.

IBETA (Y OR N)
BLADE FLAPPING DOF

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 
*                                                                  *
*   BLADE MUST HAVE AT LEAST ONE OF FLAP, LAG, OR PITCH DOF        *
*                                                                  *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 
```

ENTER 1 Y OR N VALUE

(Enter Y [yes] if blade flap degree of freedom is elected, else N [no]; at least one of IBETA, IZETA, or ITHET must equal yes)

IZETA (Y or N)
BLADE LAG DOF
ENTER 1 Y OR N VALUE

ITHET (Y or N)
BLADE PITCH DOF
ENTER 1 Y OR N VALUE

IX (Y or N)
HUB TRAN DOF - LONG
ENTER 1 Y OR N VALUE

IY (Y or N)
HUB TRAN DOF - LAT
ENTER 1 Y OR N VALUE

IZ (Y or N)
HUB TRAN DOF - AXIAL
ENTER 1 Y OR N VALUE

IAX (Y or N)
HUB ANGL DOF - ROLL
ENTER 1 Y OR N VALUE

IAY (Y or N)
HUB ANGL DOF - PITCH
ENTER 1 Y OR N VALUE

IAZ (Y or N)
HUB ANGL DOF - YAW
ENTER 1 Y OR N VALUE

PHL (REAL)
PITCH HORN LENGTH
+ VALUE IS FWD OF FA; - IS AFT
ENTER 1 REAL VALUE

(Enter pitch horn length [inches]; positive indicates that control rod degrees of freedom will be located forward of the feathering axis, negative aft)

PHSTA (REAL)
PITCH HORN STATION
DISTANCE OF STATION FROM HINGE
ENTER 1 REAL VALUE

(Enter the distance spanwise [inches] of the pitch horn from the flap/lag hinge)

NB (INTEGER)
NUMBER OF BLADES
ENTER 1 INTEGER VALUE(S)

NBACTU (INTEGER)
ACTUAL NO. OF BLADES
ENTER 1 INTEGER VALUE(S)

(If a single-blade representation is elected, enter the actual number of blades represented)

R (REAL)
ROTOR RADIUS (IN)
ENTER 1 REAL VALUE

RPM (REAL)
ROTOR RPM
ENTER 1 REAL VALUE

IC (INTEGER)
ROTOR ROTATION
- 1 IS CLOCKWISE; + 1 IS COUNTERCLOCKWISE
ENTER 1 INTEGER VALUE(S)

(Counterclockwise ≈ + ALFZ)

PSI (REAL)
AZIMUTH OF REF BLADE
AZIMUTH OF REFERENCE BLADE AT T = Ø
ENTER 1 REAL VALUE

(Enter the azimuth [degrees] of the reference blade [blade 1] feathering axis at time, t, equal to zero)

E1 (REAL)
HINGE OFFSET
ENTER 1 REAL VALUE

(Enter the distance spanwise [inches] to the flap/lag hinge from the vertical principal axis)

CBETA (REAL)
FLAP DAMPER VALUE
ENTER 1 REAL VALUE

(lb-sec-in.^{2}/in.-deg)

KBETA (REAL)
FLAP SPRING STIFFNESS
ENTER 1 REAL VALUE

(lb-in.^{2}/in.-deg)

BPC (REAL)
PRECONE ANGLE
ENTER 1 REAL VALUE

(degrees)

CZETA (REAL)
LAG DAMPER VALUE
ENTER 1 REAL VALUE

(lb-sec-in.^{2}/in.-deg)

KZETA (REAL)
LAG SPRING STIFFNESS
ENTER 1 REAL VALUE

(lb-in.^{2}/in.-deg)

CTHET (REAL)
PITCH DAMPER VALUE
ENTER 1 REAL VALUE

(lb-sec-in.^{2}/in.-deg)

KTHET (REAL)
PITCH SPRING STIFFNESS
ENTER 1 REAL VALUE

(lb-sec-in.^{2}/in.-deg)

RCOUPC (REAL)
FLAP-LAG DAMPING COUPLING
ENTER 1 REAL VALUE

(If CBETA, CZETA > Ø, may elect flap-lag damping coupling factor, Ø $\leq$ RCOUPC $\leq$ 1)

RCOUPK (REAL)
FLAP-LAG STIFFNESS COUPLING
ENTER 1 REAL VALUE

(If KBETA, KZETA > Ø, may elect flap-lag stiffness coupling factor, Ø ≤ RCOUPK ≤ 1)

MHUB (REAL)
HUB WEIGHT (LB)
ENTER 1 REAL VALUE

(Enter the hub weight [lb])

IHUBX (REAL)
HUB MOI - REF BLADE
MOMENT OF INERTIA ABOUT REFERENCE BLADE AXIS
ENTER 1 REAL VALUE

(Enter the mass moment of inertia [lb-in.^{2}] of the hub about the reference [in-plane] axis)

IHUBY (REAL)
HUB MOI - PERPENDICULAR
MOMENT OF INERTIA ABOUT PERPENDICULAR AXIS
ENTER 1 REAL VALUE

(Enter the moment of inertia of the hub about the in-plane axis perpendicular to the reference axis)

IHUBZ (REAL)
HUB MOI - SHAFT AXIS
MOMENT OF INERTIA ABOUT SHAFT AXIS
ENTER 1 REAL VALUE

(Enter the moment of inertia of the hub about the shaft axis [vertical principal axis])

THØ (REAL)
ROOT PITCH ANGLE
ENTER 1 REAL VALUE

(Enter the blade root pitch angle [degrees]; + = nose up)

UB (Y or N)
UNIFORM BLADE
ENTER 1 Y OR N VALUE

(Enter Y if uniform blade is elected, else N, nonuniform)

If uniform blade:

UMB (REAL)
BLADE WT/UNIT LENGTH
ENTER 1 REAL VALUE

(Enter the blade weight per unit length [lb/in.])

UITH (REAL)
TOTAL FEATHERING MOI
ENTER 1 REAL VALUE

(Enter the total blade mass moment of inertia about the feathering axis [lb-in.^{2}])

UCG (REAL)
CG OFFSET
+ VALUE IS FWD; - IS AFT
ENTER 1 REAL VALUE

(Enter the CG offset chordwise from the blade feathering axis [inches])

UTHX (REAL)
TOTAL BUILT-IN TWIST
- VALUE = NOSE DOWN; + VALUE = NOSE UP
ENTER 1 REAL VALUE

(Enter the difference in built-in blade pitch between the blade root and the blade tip [degrees])

NX (INTEGER)
NO. OF BLADE STATIONS
ENTER 1 INTEGER VALUE(S)

(Enter the number of stations at which the blade geometric and mass properties are to be defined; $5 \leq NX \leq 40$)

If nonuniform blade:

NX (INTEGER)
NO OF BLADE STATIONS
ENTER 1 INTEGER VALUE(S)

X (REAL)
BLADE STATIONS
ENTER NX REAL VALUES

(Enter the blade stations [inches] at which the blade geometric and mass properties are to be defined in ascending order)

MB (REAL)
BLADE WT/UNIT LENGTH
ENTER NX REAL VALUES

(Enter the blade weight per unit length [lb/in.] for each station)

ITH (REAL)
FEATHERING MOI
ENTER NX REAL VALUES

(Enter the blade mass moment of inertia per unit length about the feathering axis [lb-in.^{2}/in.] for each station)

THX (REAL)
BUILT-IN TWIST
- VALUE = NOSE DOWN; + VALUE = NOSE UP
ENTER NX REAL VALUES

(Enter the difference in built-in blade pitch between the blade root and each station [degrees])

CG (REAL)
CG OFFSET
+ VALUE IS FWD; - IS AFT
ENTER NX REAL VALUES

3.1.3 <u>CRE3 - Rotor, Elastic Blades</u>. CRE3 is a rotor system representation with elastic blade degrees of freedom defined in a rotating coordinate system coupled to hub translation and rotation degrees of freedom defined in a nonrotating coordinate system.

3.1.3.1 <u>Primary Features</u>

<u>Elastic Blades</u> - Up to 9 identical elastic blades which may have physical degrees of freedom consisting of up to:

5 out-of-plane bending modes
3 in-plane bending modes
3 torsional modes

One rotor speed perturbation degree of freedom (angular displacement of the entire rotor about the shaft axis with respect to rotating system) may also be defined. The total number of explicit degrees of freedom, including hub degrees of freedom, cannot exceed 40, however.

All blade degrees of freedom are optional, except that at least one elastic degree of freedom is required. The blade degrees of freedom are defined in a coordinate system which rotates about the hub vertical principal axis. Mode shapes may be input or may be generated automatically as the normal modes of a nonrotating beam for given

boundary conditions. Note that user-input modes need not be orthogonal as long as they satisfy deflection and slope boundary conditions. Also, curvature (2nd derivative) and slope are required (see paragraph 2.2, DYSCO 4.1 Theoretical Manual).

Hinges With Springs and Dampers - In-plane hinge, out-of-plane hinge, and pitch bearing may be included; associated springs and dampers.

Control System Interface - The blade torsion degree of freedom may be coupled to a control system component through a rigid pitch horn.

Implicit Degrees of Freedom - In addition to the control system, up to 9 other implicit degrees of freedom may be defined along one or more blades. These implicit degrees of freedom are arbitrary and may be degrees of freedom of other components, which will result in the automatic coupling of the components. The implicit degrees of freedom are automatically formulated into linear combinations of the blade degrees of freedom.

Uniform or Nonuniform Blades - Option allowed for convenience of input.

Hub Degrees of Freedom - Up to 3 translational and 3 rotational hub degrees of freedom are defined in a nonrotating Cartesian coordinate system.

Aerodynamic Forcing - Optional use of FRAØ, FRA2, or FRA3 rotor aerodynamic force modules (see paragraph 3.2 for details).

Nonlinear Terms - The second order nonlinear terms of the component equations of motion can optionally be included.

Multiple Uses - CRE3 can be used up to 4 times in a given model.

3.1.3.2 Degrees of Freedom - The names of the system degrees of freedom are formulated automatically as follows. All are optional, except that at least one elastic degree of freedom is required.

Blade Degrees of Freedom (Rotating System) -

IP rbmØ - in-plane

OP rbmØ - out-of-plane

TOR rbmØ - torsion

ALFSrØØØ - rotor speed perturbation

Hub Degrees of Freedom (Nonrotating System) -

XHUBrØØØ - longitudinal translation

YHUBrØØØ - lateral translation

ZHUBrØØØ - vertical translation

ALFXrØØØ - roll

ALFYrØØØ - pitch

ALFZrØØØ - yaw

where r = rotor number (assigned during model formulation)

b = blade number (blade 1 is the reference blade)

m = elastic mode number

Implicit Degrees of Freedom (Rotating System) - The relationship between control rod degrees of freedom (RODRrbØØ) and blade degrees of freedom is

RODRrbØØ = PHL * TOR rb(Σm)Ø + PHSTA * OP rb(Σm)Ø

where PHL is the pitch horn length, PHSTA is the pitch horn station, and Σm represents the summation of modes of a degree of freedom at PHSTA.

The implicit relationship is automatically established if blade torsion is a degree of freedom and PHL ≠ Ø. This is used to automatically couple the rotor component to an associated control system component (CCEØ or CCE1).

Similarly, the relationship between an implicit degree of freedom and the degree of freedom of a single blade is:

IMPLØØØØ = DIST * TOR rb (Σm)Ø + XSTA * OP rb(Σm)Ø

where IMPLØØØØ is some implicit degree of freedom, DIST is the distance to the blade elastic center, and XSTA is the blade station of the implicit degree of freedom. The positive orientations of the component degrees of freedom are presented in Figure 15. Only the literal portions of the degree of freedom names are shown. ψ is the azimuth angle of the reference blade (blade 1).

3.1.3.3 Input, CRE3 - The CRE3 input prompts are shown below in capital letters. Input prompt format is:

```
VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS (FORTRAN READ FORMAT)
response
```

Only required input (based on previous input) is requested.

```
JV (Y OR N)
INPLANE DOF
ENTER 1 Y OR N VALUE
```

(Enter Y [yes] if blade in-plane degree of freedom is elected, else N [no]; at least one of JV, JW, or JP must equal yes)

```
JW (Y OR N)
OUTPLANE DOF
ENTER 1 Y OR N VALUE

JP (Y OR N)
TORSION DOF
ENTER 1 Y OR N VALUE

JS (Y OR N)
SHAFT PERTURBED DOF
ENTER 1 Y OR N VALUE

JX (Y OR N)
XHUB (LONG) DOF
ENTER 1 Y OR N VALUE

JY (Y OR N)
YHUB (LAT) DOF
ENTER 1 Y OR N VALUE
```

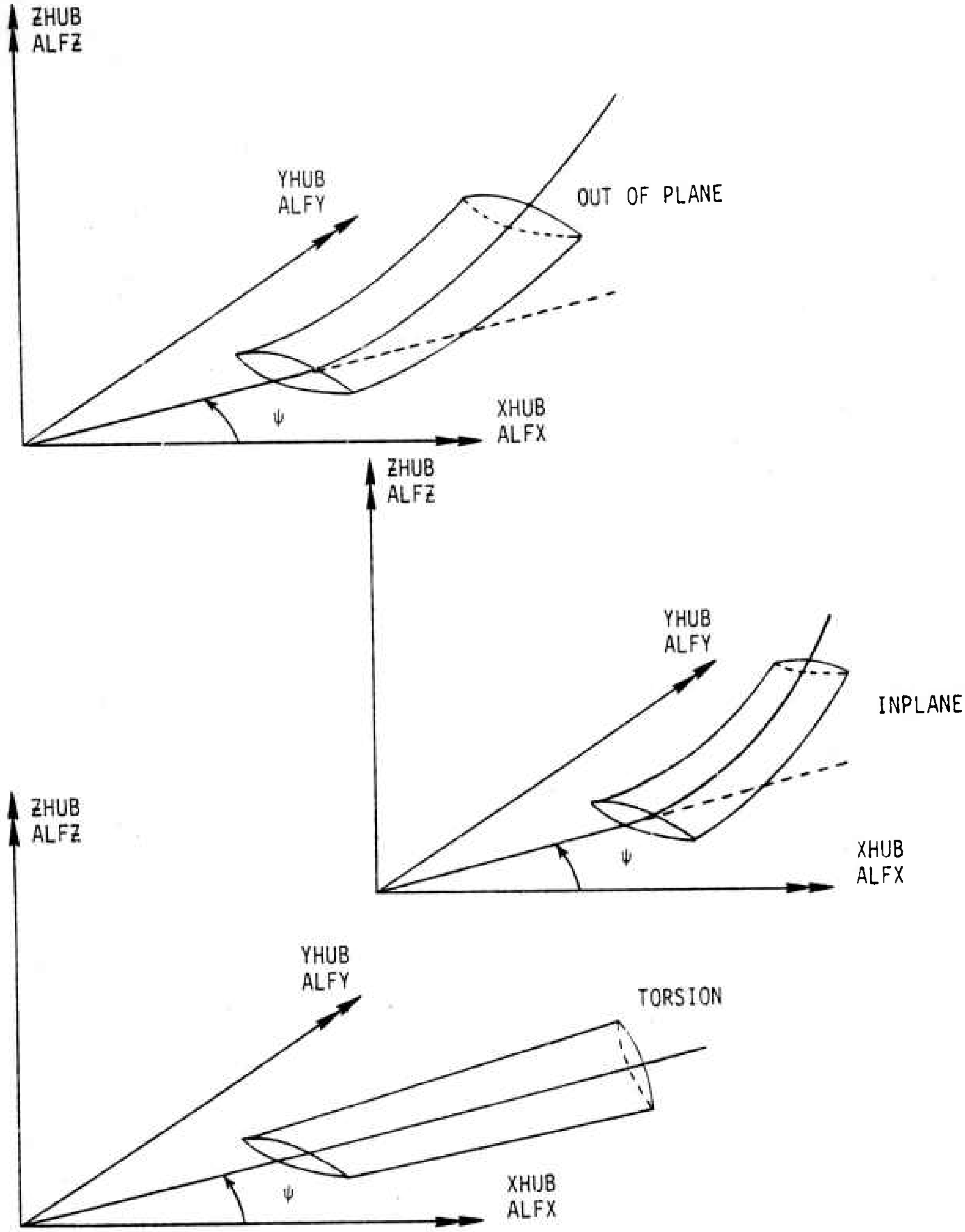


Figure 15. CRE3 Degrees of Freedom.

JZ (Y OR N)
ZHUB (AXIAL) DOF
ENTER 1 Y OR N VALUE

JAX (Y OR N)
ALFX (ROLL) DOF
ENTER 1 Y OR N VALUE

JAY (Y OR N)
ALFY (PTCH) DOF
ENTER 1 Y OR N VALUE

JAZ (Y OR N)
ALFZ (YAW) DOF
ENTER 1 Y OR N VALUE

NV (INTEGER)
NO. OF INPLANE MODES
ENTER 1 INTEGER VALUE(S)

(Enter the number of in-plane modes; $1 \leq NV \leq 3$)

NW (INTEGER)
NO. OF OUTPLANE MODES
ENTER 1 INTEGER VALUE(S)

(Enter the number of out-of-plane modes; $1 \leq NW \leq 5$)

NP (INTEGER)
NO. OF TORSION MODES
ENTER 1 INTEGER VALUE(S)

(Enter the number of torsion modes; $1 \leq NP \leq 3$)

NB (INTEGER)
NO. OF BLADES
ENTER 1 INTEGER VALUE(S)
($1 \leq NB \leq 9$)

NBACTU (INTEGER)
ACTUAL NO. OF BLADES
ENTER 1 INTEGER VALUE(S)

(If a single-blade representation is elected, enter the actual number of blades represented; $1 \leq NBACTU \leq 9$)

PHL (REAL)
PTCH HORN LENGTH (IN)
(+ FWD OF FA)
ENTER 1 REAL VALUE

(Enter pitch horn length [inches]; positive indicates that control rod degrees of freedom will be located forward of the feathering axis, negative aft; note the blade feathering axis is assumed to coincide with the elastic axis)

PHSTA (REAL)
PTCH HORN STA (IN)
ENTER 1 REAL VALUE

(Enter the distance [inches] to the pitch horn from the hub vertical principal axis [blade station of the pitch horn]; PHSTA $\geq \emptyset$)

NX (INTEGER)
NO. OF STATIONS
ENTER 1 INTEGER VALUE(S)

(Enter the number of stations at which the blade geometric and physical properties and mode shapes are to be defined; $12 \leq NX \leq 40$)

ITYP (INTEGER)
MODE INPUT 1 OR 2
1 = MODE SHAPE INPUT BY USER
2 = MODE SHAPE GENERATED AUTOMATICALLY BASED ON UNIFORM NONROTATING BEAM
ENTER 1 INTEGER VALUE(S)

X (REAL)
STATIONS
ENTER NX REAL VALUE(S)

(Enter the blade definition stations [inches] in ascending order; $X \geq \emptyset$)

NIP (INTEGER)
INPLANE HINGE STA
ENTER 1 INTEGER VALUE(S)

(Enter the index of the blade station, X(I), at which the in-plane hinge is located; $1 \leq NIP \leq 5$)

NOP (INTEGER)
OUTPLANE HINGE STA
ENTER 1 INTEGER VALUE(S)
($1 \leq NOP \leq 5$)

NTOR (INTEGER)
PTCH BEARING STA
ENTER 1 INTEGER VALUE(S)
($1 \leq NTOR \leq 5$)

VPP (REAL)
2ND DERIVATIVE OF IP
MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NV/NX REAL VALUE(S)

(Enter the second derivatives of the in-plane mode shape at the nth blade station/enter the second derivative of the nth in-plane mode shape)

VP (REAL)
1ST DERIVATIVE OF IP
MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NV/NX REAL VALUE(S)

V (REAL)
INPLANE MODE SHAPES
PREPARE TO ENTER ROW/COLUMN n
ENTER NV/NX REAL VALUE(S)

(Enter the in-plane mode shapes at the nth blade station/enter the nth in-plane mode shape)

WPP (REAL)
2ND DERIVATIVE OF OP
MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NW/NX REAL VALUE(S)

(See VPP, above)

WP (REAL)
1ST DERIVATIVE OF OP
MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NW/NX REAL VALUE(S)

W (REAL)
OUTPLANE MODE SHAPES
PREPARE TO ENTER ROW/COLUMN n
ENTER NW/NX REAL VALUE(S)

(See V, above)

PHPP (REAL)
2ND DERIVATIVE OF
TORSION MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NP/NX REAL VALUE(S)

(See VPP, above)

PHP (REAL)
1ST DERIVATIVE OF
TORSION MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NP/NX REAL VALUE(S)

PH (REAL)
TORSION MODE SHAPES
PREPARE TO ENTER ROW/COLUMN n
ENTER NP/NX REAL VALUE(S)

(See V, above)

CIPP (REAL)
IP MODAL DAMPING
(LBF-SEC/IN)
ENTER NV REAL VALUE(S)

(Enter the modal damping for each in-plane mode)

COPP (REAL)
OP MODAL DAMPING
(LBF-SEC/IN)
ENTER NW REAL VALUE(S)

CTORR (REAL)
TORSION MODAL DAMPING
(IN-LBF-SEC/DEG)
ENTER NP REAL VALUE(S)

IBIP (INTEGER)
IP BC 1 OR 2
1 = PIN-FREE 2 = CANT-FREE
ENTER 1 INTEGER VALUE(S)

(Select the in-plane boundary conditions for automatic mode shape generation, ITYP = 2)

IBOP (INTEGER)
OP BC 1 OR 2
1 = PIN-FREE 2 = CANT-FREE
ENTER 1 INTEGER VALUE(S)

IBTO (INTEGER)
TORSION BC 1 OR 2
1 = FREE-FREE 2 = FIXED-FREE
ENTER 1 INTEGER VALUE(S)

NI (INTEGER)
NO. OF IMPLICIT DOFS
ENTER 1 INTEGER VALUE(S)
($\emptyset \leq$ NI ≤ 9)

NIDOF (DOF)
IMPLICIT DOF NAME
ENTER NI DOF NAMES (A4,I4) ONE PER LINE

XSTA (REAL)
STA FOR EACH IMP DOF
(IN)
ENTER NI REAL VALUE(S)

(Enter the implicit degree of freedom blade stations)

DIST (REAL)
DISTANCE TO EC
+ FWD OF EC
ENTER NI REAL VALUE(S)

(Enter the distance [inches] to the blade elastic center for each implicit degree of freedom; positive indicates that the degree of freedom is located forward of the elastic center)

IBN (INTEGER)
BLADE NO. FOR IDOFS
ENTER NI INTEGER VALUE(S)
(Enter the blade number for each implicit degree of freedom)

KIP (REAL)
IP SPRING RATE
(IN-LBF/DEG)
ENTER 1 REAL VALUE

(Enter the spring rate for the in-plane hinge)

CIP (REAL)
IP DAMPING RATE
(IN-LBF-SEC/DEG)
ENTER 1 REAL VALUE

(Enter the damping rate for the in-plane hinge)

KOP (REAL)
OP SPRING RATE
(IN-LBF/DEG)
ENTER 1 REAL VALUE

(Enter the spring rate for the out-of-plane hinge)

COP (REAL)
OP DAMPING RATE
(IN-LBF-SEC/DEG)
ENTER 1 REAL VALUE

(Enter the damping rate for the out-of-plane hinge)

KTOR (REAL)
TORSION SPRING RATE
(IN-LBF/DEG)
ENTER 1 REAL VALUE

(Enter the spring rate for the pitch bearing; note that a control rod acts as a torsional spring)

CTOR (REAL)
TORSION DAMPING RATE
(IN-LBF-SEC/DEG)
ENTER 1 REAL VALUE

(Enter the damping rate for the pitch bearing; note that a control rod can act [depending on CCEØ, CCE1] as a torsional damper)

OM (REAL)
RPM
ENTER 1 REAL VALUE

(Enter the rotor rpm)

IC (INTEGER)
DIRECTION OF ROTATION
1 = COUNTERCLOCKWISE -1 = CLOCKWISE
ENTER 1 INTEGER VALUE(S)

(Counterclockwise ≈ + ALFZ)

PSIØ (REAL)
AZIMUTH OF REF BLADE
AT T = Ø (DEG)
ENTER 1 REAL VALUE

(Enter the azimuth of the reference blade [blade 1] feathering axis at time, t, equal to zero; -360° ≤ PSIØ ≤ 360°)

BPCØ (REAL)
PRECONE ANGLE (DEG)
ENTER 1 REAL VALUE

MHUB (REAL)
HUB WEIGHT (LB)
ENTER 1 REAL VALUE

(MHUB ≥ Ø)

IHUBX (REAL)
HUB MOI ABOUT X-AXIS
(LB-IN**2)
ENTER 1 REAL VALUE

IHUBY (REAL)
HUB MOI ABOUT Y-AXIS
(LB-IN**2)
ENTER 1 REAL VALUE

IHUBZ (REAL)
HUB MOI ABOUT Z-AXIS
(LB-IN**2)
ENTER 1 REAL VALUE

THØ (REAL)
ROOT PTCH ANG (DEG)
ENTER 1 REAL VALUE

(Enter the blade root pitch angle; + = nose up)

NONLIN (Y OR N)
NONLIN TERMS
ENTER 1 Y OR N VALUE

(Enter Y if the second order nonlinear terms of the component equations of motion are to be included)

IU (Y OR N)
UNIFORM BLADE
ENTER 1 Y OR N VALUE

(Enter Y if uniform blade is elected, else N, nonuniform)

If uniform blade:

MØ (REAL)
UNIFORM MASS DENSITY
(LB/IN)
ENTER 1 REAL VALUE

(Enter the uniform blade weight per unit length; MØ $\geq$ Ø)

SEØ (REAL)
UNIFORM CG OFFSET
FROM EA (IN)
ENTER 1 REAL VALUE

(Enter the uniform CG offset from the elastic axis; positive - forward)

SEAØ (REAL)
UNIFORM AC OFFSET
FROM EA (+ FWD EA) (IN)
ENTER 1 REAL VALUE

(Enter the uniform area centroid offset from the elastic axis; positive - forward)

KM1Ø (REAL)
UNIFORM MASS ROG
ABOUT CHORDWISE AXIS
IN BEAMWISE DIRECTION (IN)
ENTER 1 REAL VALUE

(Enter the uniform mass radius of gyration in the beamwise [perpendicular to spanwise-chordwise plane] direction; KM1Ø $\geq$ Ø)

KM2Ø (REAL)
UNIFORM MASS ROG
ABOUT BEAMWISE AXIS
IN CHORDWISE DIRECTION (IN)
ENTER 1 REAL VALUE

(KM2Ø $\geq$ Ø)

KAØ (REAL)
UNIFORM AREA ROG
ABOUT SPANWISE AXIS
OF CROSS SECTION (IN)
ENTER 1 REAL VALUE
(Enter the uniform area radius of gyration in the chordwise-beamwise plane [KAØ = $\sqrt{Y^2 + Z^2}$]; KAØ $\geq$ Ø)

THPØ (REAL)
UNIFORM PRETWIST RATE
(DEG/IN)
ENTER 1 REAL VALUE

(Enter the uniform built-in twist per unit length)

EIYØ (REAL)
UNIFORM CHORDWISE EI
*1ØE - Ø6 (LBF*IN**2)
ENTER 1 REAL VALUE

(Enter the uniform chordwise [in-plane bending] stiffness; EIYØ $\geq$ Ø)

EAØ (REAL)
UNIFORM SECTION EA
*1ØE - Ø6
(LBF)
ENTER 1 REAL VALUE

(Enter the uniform spanwise elongation stiffness; EAØ $\geq$ Ø)

EIZØ (REAL)
UNIFORM BEAMWISE EI
*1ØE - Ø6 (LBF*IN**2)
ENTER 1 REAL VALUE

(Enter the uniform beamwise [out-of-plane bending] stiffness; EIZØ $\geq$ Ø)

GJØ (REAL)
UNIFORM SECTION GJ
*1ØE - Ø6 (LBF*IN**2)
ENTER 1 REAL VALUE

(Enter the uniform torsional stiffness; GJØ $\geq$ Ø)

EB1Ø (REAL)
UNIFORM CROSS SEC
INTEGRAL
(IN**6)
ENTER 1 REAL VALUE

(Area integral of cross section with respect to local principal axes [see Volume I, Theoretical Manual])

EB2Ø (REAL)
UNIFORM CROSS SEC
INTEGRAL
(IN**5)
ENTER 1 REAL VALUE

EC1Ø (REAL)
UNIFORM CROSS SEC
INTEGRAL
(IN**6)
ENTER 1 REAL VALUE

EC1STAØ (REAL)
UNIFORM CROSS SEC
INTEGRAL
(IN**5)
ENTER 1 REAL VALUE

If nonuniform blade:

M (REAL)
WT PER UNIT LENGTH
(LB/IN)
ENTER NX REAL VALUE(S)

(Enter the blade weight per unit length at each blade station; M(I) $\geq$ Ø)

SE (REAL)
CG OFFSET FROM EA (IN)
ENTER NX REAL VALUE(S)

(Enter the CG offset from the elastic axis at each blade station; positive - forward)

SEA (REAL)
AREA CENTROID OFFSET
FROM EA (+ FWD EA) (IN)
ENTER NX REAL VALUE(S)

(Enter the area centroid offset from the elastic axis at each blade station; positive - forward)

KM1 (REAL)
MASS ROG ABOUT LOCAL
CHORDWISE AXIS
IN BEAMWISE DIRECTION (IN)
ENTER NX REAL VALUE(S)

(Enter the mass radius of gyration in the beamwise [perpendicular to spanwise-chordwise plane] direction at each blade station; KM1(I) $\geq$ Ø)

KM2 (REAL)
MASS ROG ABOUT LOCAL
BEAMWISE AXIS
IN CHORDWISE DIRECTION (IN)
ENTER NX REAL VALUE(S)
(KM2(I) $\geq$ Ø)

KA (REAL)
AREA ROG ABOUT LOCAL
SPANWISE AXIS
OF CROSS SECTION (IN)
ENTER NX REAL VALUE(S)

(Enter the area radius of gyration in the chordwise-beamwise plane [KA = $\sqrt{Y^2 + Z^2}$] at each blade station; KA(I) $\geq$ Ø)

THP (REAL)
PRETWIST RATE (DEG/IN)
ENTER NX REAL VALUE(S)

(Enter the built-in twist per unit length at each blade section)

EIY (REAL)
CHORDWISE EI*1ØE - Ø6
(LBF*IN**2)
ENTER NX REAL VALUE(S)

(Enter the chordwise [in-plane bending] stiffness at each blade station; EIY(I) $\geq$ Ø)

EA (REAL)
SECTION EA * 1ØE - Ø6
(LBF)
ENTER NX REAL VALUE(S)

(Enter the spanwise elongation stiffness at each blade station; EA(I) $\geq$ Ø)

EIZ (REAL)
BEAMWISE EI*1ØE - Ø6
(LBF*IN**2)
ENTER NX REAL VALUE(S)

(Enter the beamwise [out-of-plane bending] stiffness at each blade station; EIZ(I) $\geq$ Ø)

GJ (REAL)
SECTION GJ*1ØE - Ø6
(LBF*IN**2)
ENTER NX REAL VALUE(S)

(Enter the torsional stiffness at each blade station; GJ(I) $\geq$ Ø)

EB1 (REAL)
CROSS SEC INTEGRAL
(IN**6)
ENTER NX REAL VALUE(S)

(Area integral of cross section at each blade station with respect to local principal axes [see Volume I, Theoretical Manual])

EB2 (REAL)
CROSS SEC INTEGRAL
(IN**5)
ENTER NX REAL VALUE(S)

EC1 (REAL)
CROSS SEC INTEGRAL
(IN**6)
ENTER NX REAL VALUE(S)

EC1STA (REAL)
CROSS SEC INTEGRAL
(IN**5)
ENTER NX REAL VALUE(S)

JIL (Y OR N)
INTERNAL LOADS
ENTER 1 Y OR N VALUE

(Enter Y if internal loads [blade moments] are to be calculated from time history; see SII3)

NXIL (INTEGER)
NO. OF STATIONS
ENTER 1 INTEGER VALUE

(Enter the number of blade stations at which blade moments will be calculated; $1 \leq NXIL \leq NX$)

INDIL (INTEGER)
STATION INDICES
ENTER NXIL INTEGER VALUES

(Enter the indices of the NXIL blade stations in ascending order; $1 \leq INDIL(I) \leq NX$)

JIPIL (Y OR N)
INPLANE MOMENTS
ENTER 1 Y OR N VALUE

(Enter Y if in-plane moments are to be calculated)

JOPIL (Y OR N)
OUTPLANE MOMENTS
ENTER 1 Y OR N VALUE

(Enter Y if out-of-plane moments are to be calculated)

JTORIL (Y OR N)
TWIST MOMENTS
ENTER 1 Y OR N VALUE
(Enter Y if twist moments due to shear are to be calculated)

3.1.4 <u>CCE1 - Control System, Elastic Rods</u>. CCE1 represents a rotor control system consisting of damped elastic control rods coupled to the rotor blades and a swash plate.

3.1.4.1 <u>Primary Features</u> -

<u>Damped Elastic Control Rods</u> - A control rod interfaces each blade of the associated rotor at the pitch horn (rotating system; see paragraph 3.1.2.2, Implicit Degrees of Freedom) and interfaces the swash plate at the same azimuth angle as the blade.

<u>Collective and Cyclic Swash Plate</u> - Swash plate vertical (collective) displacement, roll, and pitch degrees of freedom are defined with respect to the rotor hub principal axes. A spring and a damper may be specified for each swash plate degree of freedom.

Model Formulation Constraints - During model formulation, CCE1 must be added after the corresponding rotor and must be given the same component number. Force modules are not accepted by CCE1.

3.1.4.2 Degrees of Freedom - All degree of freedom names are formulated automatically as follows:

Rotating System -

RODRrbØØ - displacement of rod end at pitch horn

Nonrotating System -

ZSPTrØØØ - swash plate vertical (collective) displacement
ASPXrØØØ - swash plate roll
ASPYrØØØ - swash plate pitch

where r = rotor number (assigned during model formulation)
b = blade number

The positive orientations of the component degrees of freedom are presented in Figure 16. Only the literal portions of the degree of freedom names are shown. ψ is the azimuth angle of the reference blade (blade 1).

3.1.4.3 Input, CCE1 - The CCE1 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response

Only required input (based on previous input) is requested.

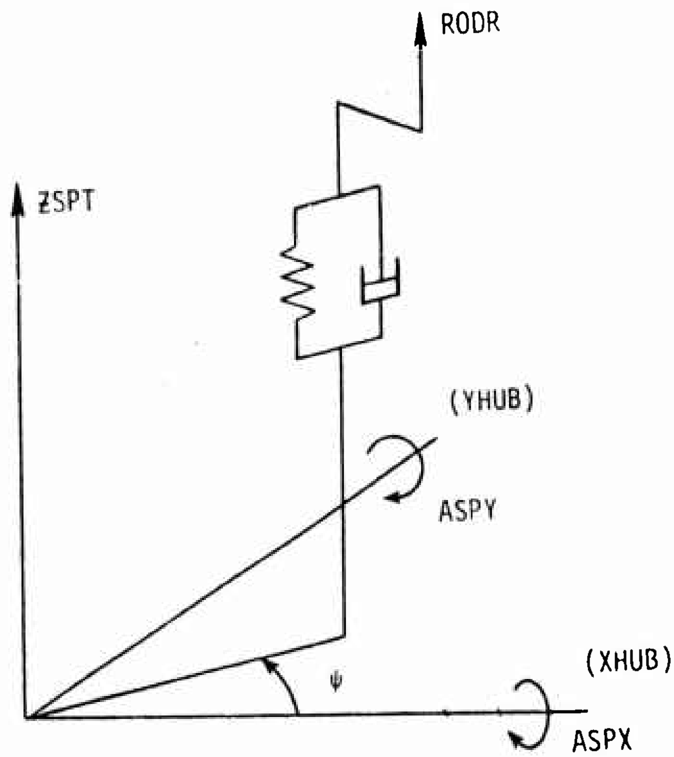


Figure 16. CCE1 Degrees of Freedom.

MSP (REAL)
SWASH PLATE WEIGHT
(LB)
ENTER 1 REAL VALUE

(Enter the swash plate weight [lb]; MSP > Ø)

ISP (REAL)
MOI ABOUT DIAMETER
(LB-IN SQ)
ENTER 1 REAL VALUE

(Enter the moment of inertia of the swash plate about its diameter [lb-in.^{2}]; ISP > Ø)

CSCOL (REAL)
COLLECTIVE DAMPING
(LB-SEC/IN)
ENTER 1 REAL VALUE

(CSCOL ≥ Ø)

CSCYC (REAL)
CYCLIC DAMPING
(LB-SEC-IN/DEG)
ENTER 1 REAL VALUE

(The damping coefficient is the same for swash plate roll and pitch; CSCYC ≥ Ø)

KSCOL (REAL)
COLLECTIVE STIFFNESS
(LB/IN)
ENTER 1 REAL VALUE

(KSCOL ≥ Ø)

KSCYC (REAL)
CYCLIC STIFFNESS
(LB-IN/DEG)
ENTER 1 REAL VALUE

(The stiffness coefficient is the same for swash plate roll and pitch; KSCYC ≥ Ø)

LROD (REAL)
AXIS-CONTROL ROD
DISTANCE
(IN)
ENTER 1 REAL VALUE

(Enter the distance to the control rods from the vertical principal axis; LROD ≥ Ø)

CROD (REAL)
CONTROL ROD DAMPING
(LB-SEC/IN)
ENTER 1 REAL VALUE

(CROD $\geq$ Ø)

KROD (REAL)
CONTROL ROD STIFFNESS
(LB/IN)
ENTER 1 REAL VALUE

(KROD $\geq$ Ø)

3.1.5 CCEØ - Control System, Elastic Rods. CCEØ represents a rotor control system consisting of undamped elastic control rods coupled to the rotor blades.

3.1.5.1 Primary Features -

Undamped Elastic Control Rods - A control rod interfaces each blade of the associated rotor at the pitch horn (rotating system; see paragraph 3.1.2.2, Implicit Degrees of Freedom).

Model Formulation Constraints - During model formulation, CCEØ must be added after the corresponding rotor and must be given the same component number. Force modules are not accepted by CCEØ.

3.1.5.2 Degrees of Freedom - All degree of freedom names are formulated automatically as follows:

Rotating System -

RODRrbØØ - displacement of rod end at pitch horn

where r = rotor number (assigned during model formulation)
b = blade number

The positive orientations of the component degrees of freedom are the same as for CCE1 (Figure 16), except that there are no swash plate degrees of freedom.

3.1.5.3 Input, CCEØ - The CCEØ input prompt is shown below in capital letters. Input prompt format is:

```
VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response

KROD (REAL)
CONTROL ROD STIFFNESS
(LB/IN)
ENTER 1 REAL VALUE
(KROD ≥ Ø)
```

3.1.6 CSF1 - Structure, Finite Element. CSF1 represents an arbitrary, linear, constant coefficient set of second-order differential equations of the form

$$M\ddot{X} + C\dot{X} + KX = F$$

where M, C, and K are coefficient matrices, F is a vector, and x represents the degrees of freedom of the system. The degrees of freedom are general and may represent displacements or rotations.

3.1.6.1 Primary Features -

Constant Coefficients - Coefficient matrices M, C, and K, and vector F may be input for up to 40 degrees of freedom.

Coupling - Degrees of freedom may be degrees of freedom of other components, which will result in the automatic coupling of the components. Through this mechanism, a CSF1 data set may be used to modify the constant coefficients of the system equations of a model.

Global Reference - In conjunction with the global reference option (see paragraph 3.4), the user may specify transformations to the local degree of freedom coordinate systems of lumped parameter systems.

Sinusoidal Force Input - CSF1 may be used in conjunction with FSS1 in order to apply a sinusoidally varying forcing function to a model (see paragraph 3.2).

Multiple Uses - CSF1 can be used up to 20 times in a given model.

3.1.6.2 Degrees of Freedom - All degree of freedom names must be supplied by the user.

3.1.6.3 Input, CSF1 - The CSF1 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS (FORTRAN READ FORMAT)
response

Only required input (based on previous input) is requested.

NCDF (INTEGER)
NUMBER OF DOF
ENTER 1 INTEGER VALUE(S)

(Enter the number of degrees of freedom; $\emptyset$ < NCDF $\leq$ 40)

CDFLI (DOF)
DOF NAMES
ENTER NCDF DOF NAMES (A4, I4) ONE PER LINE

CM (REAL)
MASS MATRIX VALUES
TYPE MATRIX
($\emptyset$ = NULL), (1 = DIAGONAL), (2 = SYMMETRIC), (3 = GENERAL)

if 1:
INPUT NCDF DIAGONAL REAL VALUES

if 2:
PREPARE TO ENTER ROW n
ENTER n REAL VALUES

if 3:
PREPARE TO ENTER ROW n
ENTER NCDF REAL VALUES

([M]; mass [lb-sec^2/in.]; mass moment of inertia [lb-sec^2-in.^{2}/in.]

CC (REAL)
DAMPING MATRIX VALUES
TYPE MATRIX
(Ø = NULL), (1 = DIAGONAL), (2 = SYMMETRIC), (3 = GENERAL)

([C]; damping [lb-sec/in.]; torsional damping [lb-sec-in.^{2}/in.-rad]

CK (REAL)
STIFFNESS MATRIX
VALUES
TYPE MATRIX
(Ø = NULL), (1 = DIAGONAL), (2 = SYMMETRIC), (3 = GENERAL)

([K]; stiffness [lb/in.]; torsional stiffness [lb-in.^{2}/in.-rad]

CF (REAL)
FORCE VECTOR VALUES
ENTER NCDF REAL VALUE(S)

([F]; force [lb]; moment [lb-in.]

IGR (Y OR N)
GLOBAL REFERENCE
ENTER 1 Y OR N VALUE

(Enter Y [yes] if global reference is to be used, else N [no])

LDC (REAL)
LOCAL DOF VECTORS
IN TERMS OF COMPONENT REFERENCE COORDINATE SYSTEM
PREPARE TO ENTER ROW/COLUMN n
ENTER 3/NCDF REAL VALUES

(Enter the local degree of freedom vectors in terms of the component coordinate system)

3.1.7 CES1 - Elastic Stop (Nonlinear Spring). CES1 represents multiple spring-damper elements which have zero values over specified gap distances. Elements may be attached to ground or to a specified base degree of freedom.

3.1.7.1 Primary Features -

Spring-Damper Elements - Up to 40 identical spring-damper systems with associated degrees of freedom.

Multiple Uses - CES1 can be used up to 4 times in a given model.

3.1.7.2 Degrees of Freedom - All degrees of freedom are optional and supplied by the user. The positive orientations of the component degrees of freedom and the arrangement of spring-damper elements are presented in Figure 17.

3.1.7.3 Input, CES1 - The CES1 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS (FORTRAN READ FORMAT)
response

Only required input (based on previous input) is requested.

MCDF (INTEGER)
NUMBER OF DOF-EXCEPT
BASE
ENTER 1 INTEGER VALUE(S)

(Enter the number of degrees of freedom, excluding the base degree of freedom; $\emptyset$ < MCDF $\leq$ 40)

CDFLI (DOF)
DOF NAMES
ENTER MCDF DOF NAMES (A4, I4) ONE PER LINE

BASE (Y OR N)
EXISTENCE OF BASE DOF
ENTER 1 Y OR N VALUE

(Enter Y if base degree of freedom is elected, else N)

CDFLBI (DOF)
BASE DOF NAME
ENTER 1 DOF NAME (A4, I4)

C1 (REAL)
UPPER DAMPING COEFF
ENTER 1 REAL VALUE

(The damping coefficient is the same for all upper elements in a given usage; damping [lb-sec/in.]; torsional damping [lb-sec-in.^{2}/in.-rad]; C1 $\geq \emptyset$)

C2 (REAL)
LOWER DAMPING COEFF
ENTER 1 REAL VALUE

(The damping coefficient is the same for all lower elements in a given usage; C2 $\geq \emptyset$)

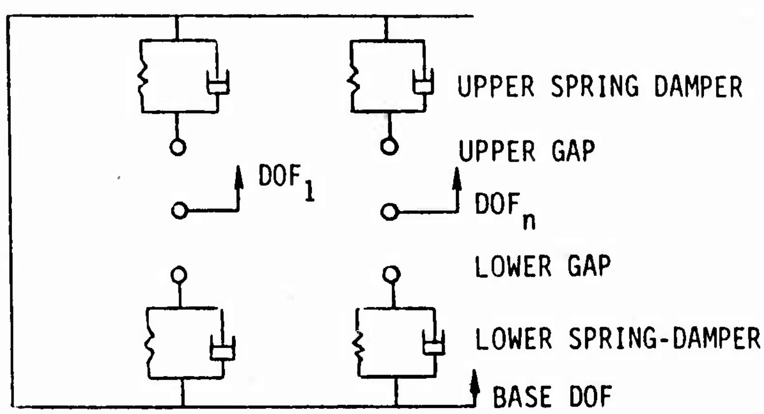


Figure 17. CES1 Degrees of Freedom.

K1 (REAL)
UPPER SPRING COEFF
ENTER 1 REAL VALUE

(The stiffness coefficient is the same for all upper elements in a given usage; stiffness [lb/in.]; torsional stiffness [lb-in.^{2}/in.-rad]; K1 $\geq$ Ø)

K2 (REAL)
LOWER SPRING COEFF
ENTER 1 REAL VALUE

(The stiffness coefficient is the same for all lower elements in a given usage; K2 $\geq$ Ø)

DELT1 (REAL)
UPPER GAP SIZE
ENTER 1 REAL VALUE

(The gap is the same for all upper elements in a given usage; displacement [in.]; angular displacement [rad], DELT1 $\geq$ Ø)

DELT2 (REAL)
LOWER GAP SIZE
ENTER 1 REAL VALUE

(The gap is the same for all lower elements in a given usage; DELT2 $\geq$ Ø)

3.1.8 CLC1 - Arbitrary Linear Constraints. CLC1 is used to apply arbitrary linear constraints to the degrees of freedom of a system. The constraint equations are of the form

{IDOF} = [T]{DOF}

where IDOF are existing implicit degrees of freedom of the system which are replaced by linear combinations of DOF, arbitrary degrees of freedom, when the system equations are formed.

3.1.8.1 Primary Features -

Implicit Degrees of Freedom - Up to 20 implicit degrees of freedom. An implicit degree of freedom may only be designated as such once in a given model. This includes relationships automatically formed in

other technical modules. If an implicit degree of freedom is specified more than once, the last relationship in which it occurs is used.

Arbitrary Degrees of Freedom - Linear combinations of up to 40 degrees of freedom may be formulated to replace implicit degrees of freedom.

Coupling - Rigid, pinned, and elastic coupling of components may be accomplished. A degree of freedom may be eliminated by setting it to zero.

Multiple Uses - CLC1 can be used up to 20 times with a maximum of 100 implicit coefficients in a single usage and 60 implicit degrees of freedom and 1,000 implicit coefficients in a model.

3.1.8.2 Degrees of Freedom - All degree of freedom names must be supplied by the user.

3.1.8.3 Input, CLC1 - The CLC1 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS (FORTRAN READ FORMAT)

response

Only required input (based on previous input) is requested.

NCDF (INTEGER)
NUMBER OF DOF
ENTER 1 INTEGER VALUE(S)

(Enter the number of arbitrary degrees of freedom; $\emptyset < NCDF \leq 40$)

CDFLI (DOF)
DOF NAMES
ENTER NCDF DOF NAMES (A4, I4) ONE PER LINE

(Enter the names of the arbitrary degrees of freedom)

NCIDF (INTEGER)
OF CONSTRAINT EQNS
ENTER 1 INTEGER VALUE(S)

(Enter the number of constraint equations [number of implicit degrees of freedom]; Ø < NCIDF ≤ 20)

CIDFLI (DOF)
IMPLICIT DOF NAMES
ENTER NCIDF DOF NAMES (A4, I4) ONE PER LINE

COEF (REAL)
COEFFICIENT MATRIX
TYPE MATRIX
(Ø = NULL), (1 = DIAGONAL), (2 = SYMMETRIC), (3 = GENERAL)

(Enter the coefficient matrix T; the number of rows in the matrix is equal to the number of constraint equations [implicit degrees of freedom], NCIDF, and the number of columns is equal to the number of arbitrary degrees of freedom, NCDF)

if 1:

INPUT NCIDF = NCDF DIAGONAL REAL VALUES

if 2:

PREPARE TO ENTER ROW n
ENTER n REAL VALUES

if 3:

PREPARE TO ENTER ROW n
ENTER NCDF REAL VALUES

3.1.9 <u>CFM3 - Fuselage, Modal (3-d)</u>. CFM3 is a general modal representation for 3-dimensional elastic structures.

3.1.9.1 <u>Primary Features</u> -

<u>Rigid Body Modes</u> - Up to three translational and three rotational rigid body degrees of freedom defined at the component center of gravity.

<u>Elastic Modes</u> - Up to 20 normal modes. For each mode up to three displacements and three rotations can be defined at up to 40 node points in a 3-dimensional coordinate system.

Implicit Degrees of Freedom - Up to three translational and three rotational implicit degrees of freedom can be defined at each node point. These implicit degrees of freedom are arbitrary and can be degrees of freedom of other components, which will result in the automatic coupling of the components. Implicit degrees of freedom are automatically processed as functions of the rigid body and elastic degrees of freedom of this component when the system equations are formed.

Aerodynamic Forcing - Optional use of FFAØ for flat plate drag or FFC2 for fuselage surface aerodynamic forces (see paragraph 3.2 for details).

Multiple Uses - CFM3 can be used up to 4 times in a given model.

3.1.9.2 Degrees of Freedom - Degree of freedom names are formulated automatically as follows:

Rigid Body Degrees of Freedom -

XC sØØØ - longitudinal translation
YC sØØØ - lateral translation
ZC sØØØ - vertical translation
RL sØØØ - roll
PH sØØØ - pitch
YW sØØØ - yaw

Elastic Degrees of Freedom -

QF smØØ

Implicit Degrees of Freedom -

TRAXsØnØ - local longitudinal translation
TRAYsØnØ - local lateral translation
TRAZsØnØ - local vertical translation
ROTXsØnØ - local roll
ROTYsØnØ - local pitch
ROTZsØnØ - local yaw

where s = structure number (assigned during model formulation)

m = elastic mode number

n = node number

The positive orientations of the rigid body and implicit degrees of freedom are presented in Figure 18. The modal displacements at each node point are defined with respect to the fuselage coordinate system, and the implicit degrees of freedom are defined locally at each node.

3.1.9.3 Input, CFM3 - The CFM3 input prompts are shown below in capital letters. Input prompt format is:

```
VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
```

response

Only required input (based on previous input) is requested.

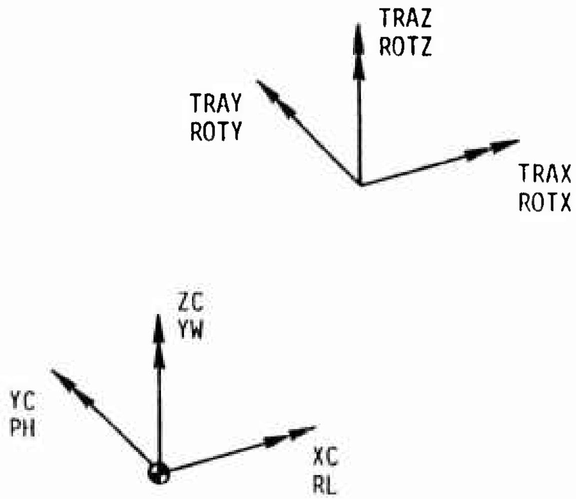


Figure 18. CFM3 Rigid Body and Implicit Degrees of Freedom.

```
RBM (Y OR N)
RIGID BODY MODES
ENTER 1 Y OR N VALUE
```

(Enter Y [yes] if rigid body modes are to be included, else N [no])

IXCG (Y OR N)
LONGITUDINAL
ENTER 1 Y OR N VALUE

(Y if longitudinal degree of freedom is elected, else N)

IYCG (Y OR N)
LATERAL
ENTER 1 Y OR N VALUE

IZCG (Y OR N)
VERTICAL
ENTER 1 Y OR N VALUE

IROLL (Y OR N)
ROLL
ENTER 1 Y OR N VALUE

IPTCH (Y OR N)
PITCH
ENTER 1 Y OR N VALUE

IYAW (Y OR N)
YAW
ENTER 1 Y OR N VALUE

CG (REAL)
XYZ CG LOCATION (IN)
ENTER 3 REAL VALUES

(Enter the coordinates of the CG [inches]; the origin of the fuselage coordinate system is defined by the user)

NS (INTEGER)
NO. OF NODAL POINTS
ENTER 1 INTEGER VALUE

($\emptyset \leq NS \leq 40$)

XYZNS (REAL)
XYZ FOR EACH NODE
XYZ REF. TO FUSELAGE SYS.
PREPARE TO ENTER ROW/COLUMN n
ENTER NS/3 REAL VALUES

(Enter the node point coordinates [inches], fuselage coordinate system)

NMODE (INTEGER)
NO. OF ELASTIC MODES
ENTER 1 INTEGER VALUE

($\emptyset \leq NMODE \leq 20$)

MXCG (Y OR N)
MODE X-COMPONENT
ENTER 1 Y OR N VALUE

(Enter Y if there are any modal displacements in the fuselage X-direction)

MYCG (Y OR N)
MODE Y-COMPONENT
ENTER 1 Y OR N VALUE

MZCG (Y OR N)
MODE Z-COMPONENT
ENTER 1 Y OR N VALUE

MROLL (Y OR N)
MODE ALFX-COMPONENT
ENTER 1 Y OR N VALUE

(Enter Y if there are any angular modal displacements with respect to the fuselage X-axis)

MPTCH (Y OR N)
MODE ALFY-COMPONENT
ENTER 1 Y OR N VALUE

MYAW (Y OR N)
MODE ALFZ-COMPONENT
ENTER 1 Y OR N VALUE

XX (REAL)
MODES X-COMPONENT
PREPARE TO ENTER ROW/COLUMN n
ENTER NS/NMODE REAL VALUES

(Enter the X-component of the nth mode shape at the NS node points/ enter the X-components of the NMODE mode shapes at the nth node point)

YY (REAL)
MODES Y-COMPONENT
PREPARE TO ENTER ROW/COLUMN n
ENTER NS/NMODE REAL VALUES

ZZ (REAL)
MODES Z-COMPONENT
PREPARE TO ENTER ROW/COLUMN n
ENTER NS/NMODE REAL VALUES

XXP (REAL)
MODES ALFX-COMPONENT
PREPARE TO ENTER ROW/COLUMN n
ENTER NS/NMODE REAL VALUES

YYP (REAL)
MODES ALFY-COMPONENT
PREPARE TO ENTER ROW/COLUMN n
ENTER NS/NMODE REAL VALUES

ZZP (REAL)
MODES ALFZ-COMPONENT
PREPARE TO ENTER ROW/COLUMN n
ENTER NS/NMODE REAL VALUES

NODOF (Y OR N)
DOF Y OR N FOR NODES
PREPARE TO ENTER ROW n
ENTER NS Y OR N VALUES (35A2)

(Rows - Y or N for TRAX TRAY TRAZ ROTX ROTY ROTZ; 6 rows, NS columns)

XYZD (REAL)
LOCAL X, Y VECTORS
IN TERMS OF FUSELAGE SYS.
PREPARE TO ENTER ROW/COLUMN n
ENTER NS/6 REAL VALUES

(Enter the X and Y vectors of the local coordinate system at each node in terms of the fuselage coordinate system [X cross Y yields the Z vector]; see Figure 19)

$$\underline{XYZD} \qquad \begin{array}{c} \\ \underline{X}' \\ \\ \\ \underline{Y}' \\ \\ \end{array} \left[\begin{array}{ccc} a_1 i & \cdots & a_{NS} i \\ b_1 j & & b_{NS} j \\ c_1 k & & c_{NS} k \\ \hdashline d_1 i & & d_{NS} i \\ e_1 j & & e_{NS} j \\ f_1 k & \cdots & f_{NS} k \end{array}\right]$$

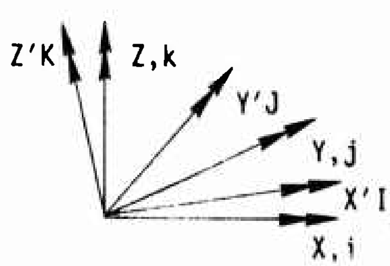


Figure 19. Local Coordinate Vectors.

MASSL (REAL)
FUSELAGE WEIGHT (LB)
ENTER 1 REAL VALUE

(Enter the fuselage weight [lb])

IMXX (REAL)
ROLL MOI(SLUG-FT (SQ))
ENTER 1 REAL VALUE

(Enter the fuselage roll mass moment of inertia [slug-ft^2])

IMYY (REAL)
PTCH MOI(SLUG-FT (SQ))
ENTER 1 REAL VALUE

IMZZ (REAL)
YAW MOI(SLUG-FT (SQ))
ENTER 1 REAL VALUE

IMXY (REAL)
XY PRODUCT OF INERT.
ENTER 1 REAL VALUE

(Enter the fuselage XY mass product of inertia [slug-ft^2])

IMYZ (REAL)
YZ PRODUCT OF INERT.
ENTER 1 REAL VALUE

IMXZ (REAL)
XZ PRODUCT OF INERT.
ENTER 1 REAL VALUE

MMS (REAL)
MODAL MASS (SLUGS)
ENTER NMODE REAL VALUES

(Enter the modal mass for each elastic mode)

MD (REAL)
MODAL DAMPING (PCT)
ENTER NMODE REAL VALUES

(Enter the percent of critical damping for each elastic mode)

FREQ (REAL)
MODAL FREQUENCY (HZ)
ENTER NMODE REAL VALUES

3.1.10 <u>CRD3 - Rotor, Damaged (Nonidentical) Blades</u>. CRD3 allows changes to be made to the physical and geometric properties of single or multiple blades of an existing elastic rotor component (CRE3) in a model.

3.1.10.1 Primary Features -

Degrees of Freedom - The CRD3 component degrees of freedom are identical to those of the associated CRE3 component. Component coefficient matrices are generated using incremental blade data supplied by the user. The component matrices are combined in the system equations and do not replace the original CRE3 component matrices.

The user is asked to select the number of blades of the existing rotor system which will be modified and the blade numbers (see paragraph 3.1.3.2). Changes for all blades selected will be identical for a given data set.

Blade Stations - The user has the option to provide a new set of stations for the modified blades or retain the stations used in the CRE3 component. This option allows finer definition of blade properties in segments of interest.

Mode Shapes - In-plane, out-of-plane, and torsion mode shapes and mode shape derivatives of the original blades can be retained for the modified blades or can be replaced by user-input modes or by automatically generated normal modes of a nonrotating beam with given boundary conditions. Note that user-input modes need not be orthogonal as long as they satisfy deflection and slope boundary conditions. In addition, slope (mode shape 1st derivative) and curvature (mode shape 2nd derivative) are required (see paragraph 2.2, DYSCO 4.1 Theoretical Manual).

Blade Properties - The user is asked to input two types of property data for the modified blades: incremental data and nonincremental data. An incremental value is the change in a property of the unmodified blades. A nonincremental value is a total value for a property, which may be different from or the same as that for the unmodified blades.

Coupling - Control system interfaces and other implicit relationships specified for the original CRE3 component are not affected by the CRD3 component.

Uniform or Nonuniform Blades - Option for convenience of blade property input.

Nonlinear Terms - The second order nonlinear terms of the component equations of motion are included automatically if they have been selected for the CRE3 component.

Model Formulation Constraints - During model formulation, CRD3 must be added after the associated CRE3 component and must be given the same component number. Rotor aerodynamic force modules (FRAØ, FRA2, FRA3) must be specified separately.

Multiple Uses - CRD3 can be used up to 4 times in a given model.

3.1.10.2 Input, CRD3 - The CRD3 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response

Only required input (based on previous input) is requested.

JV (Y OR N)
INPLANE DOF
ENTER 1 Y OF N VALUE

(Enter Y [yes] if Y for associated CRE3, else N [no])

JW (Y OR N)
OUTPLANE DOF
ENTER 1 Y OR N VALUE

(Enter Y if Y for associated CRE3, else N)

JP (Y OR N)
TORSION DOF
ENTER 1 Y OR N VALUE

(Enter Y if Y for associated CRE3, else N)

NV (INTEGER)
NO. OF INPLANE MODES
ENTER 1 INTEGER VALUE

(Enter the same number of in-plane modes as in associated CRE3)

NW (INTEGER)
NO. OF OUTPLANE MODES
ENTER 1 INTEGER VALUE

(Enter the same number of out-of-plane modes as in associated CRE3)

NP (INTEGER)
NO. OF TORSION MODES
ENTER 1 INTEGER VALUE

(Enter the same number of torsion modes as in associated CRE3)

NDB (INTEGER)
NO. OF DAMAGED
BLADES
ENTER 1 INTEGER VALUE

IDB (INTEGER)
BLADE NOS. OF
DAMAGED BLADES
ENTER NDB INTEGER VALUES

(Enter the blade numbers of the damaged blades in ascending order [see paragraph 3.1.3.2])

NX (INTEGER)
NO. OF BLADE STAS
ENTER 1 INTEGER VALUE

(Enter the number of stations at which the blade geometric and physical properties and mode shapes are to be defined; see JXD below; $12 \leq NX \leq 40$)

JXD (Y OR N)
NEW STATIONS
ENTER 1 Y OR N VALUE

(If Y, new blade stations will be input; if N, CRE3 stations will be used [NX must be the same for both components for this option])

X (REAL)
STATIONS
ENTER NX REAL VALUES

(Enter the blade definition stations [inches] in ascending order; $X \geq \emptyset$)

ITYP (INTEGER)
MODE INPUT Ø,1, OR 2
Ø = USE OLD MODE SHAPES,
1 = INPUT NEW MODE SHAPES,
2 = GENERATE NEW MODE SHAPES
ENTER 1 INTEGER VALUE

VPPD (REAL)
2ND DERIVATIVE OF IP
MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NV/NX REAL VALUES

(Enter the second derivatives of the in-plane mode shapes at the nth blade station/enter the second derivative of the nth in-plane mode shape)

VPD (REAL)
1ST DERIVATIVE OF IP
MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NV/NX REAL VALUES

VD (REAL)
INPLANE MODE SHAPES
PREPARE TO ENTER ROW/COLUMN n
ENTER NV/NX REAL VALUES

(Enter the in-plane mode shapes at the nth blade station/enter the nth in-plane mode shape)

WPPD (REAL)
2ND DERIVATIVE OF OP
MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NW/NX REAL VALUES

(See VPPD above)

WPD (REAL)
1ST DERIVATIVE OF OP
MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NW/NX REAL VALUES

WD (REAL)
OUTPLANE MODE SHAPES
PREPARE TO ENTER ROW/COLUMN n
ENTER NW/NX REAL VALUES

(See VD above)

PHPPD (REAL)
2ND DERIVATIVE OF
TORSION MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NP/NX REAL VALUES

(See VPPD above)

PHPD (REAL)
1ST DERIVATIVE OF
TORSION MODES
PREPARE TO ENTER ROW/COLUMN n
ENTER NP/NX REAL VALUES

PHD (REAL)
TORSION MODE SHAPES
PREPARE TO ENTER ROW/COLUMN n
ENTER NP/NX REAL VALUES

(See VD above)

CIPP (REAL)
IP MODAL DAMPING
INCREMENT
(LBF-SEC/IN)
ENTER NV REAL VALUES

(Enter the change [from CRE3] in modal damping for each in-plane mode)

COPP (REAL)
OP MODAL DAMPING
INCREMENT
(LBF-SEC/IN)
ENTER NW REAL VALUES

CTORR (REAL)
TORSION MODAL DAMPING
INCREMENT
(IN-LBF-SEC/DEG)
ENTER NP REAL VALUES

IBIP (INTEGER)
IP BC 1 OR 2
1 = PIN-FREE 2 = CANT-FREE
ENTER 1 INTEGER VALUE(S)

(Select the in-plane boundary conditions for automatic mode shape generation, ITYP = 2)

IBOP (INTEGER)
OP BC 1 OR 2
1 = PIN-FREE 2 = CANT-FREE
ENTER 1 INTEGER VALUE(S)

IBTO (INTEGER)
TORSION BC 1 OR 2
1 = FREE-FREE 2 = FIXED-FREE
ENTER 1 INTEGER VALUE(S)

KIP (REAL)
IP SPRING RATE
INCREMENT
(IN-LBF/DEG)
ENTER 1 REAL VALUE

(Enter the change [from CRE3] in the spring rate for the in-plane hinge)

CIP (REAL)
IP DAMPING RATE
INCREMENT
(IN-LBF-SEC/DEG)
ENTER 1 REAL VALUE

(Enter the change [from CRE3] in the damping rate for the in-plane hinge)

KOP (REAL)
OP SPRING RATE INCREMENT
(IN-LBF/DEG)
ENTER 1 REAL VALUE

COP (REAL)
OP DAMPING RATE
INCREMENT
(IN-LBF-SEC/DEG)
ENTER 1 REAL VALUE

KTOR (REAL)
TORSION SPRING RATE
INCREMENT
(IN-LBF/DEG)
ENTER 1 REAL VALUE

CTOR (REAL)
TORSION DAMPING RATE
INCREMENT
(IN-LBF-SEC/DEG)
ENTER 1 REAL VALUE

IU (Y OR N)
UNIFORM BLADE
ENTER 1 Y OR N VALUE

(Enter Y if uniform blade changes, else N, nonuniform)

If uniform blade:

MØ (REAL)
UNIFORM WT DENSITY
INCREMENT (LB/IN)
ENTER 1 REAL VALUE

(Enter the uniform change in blade weight per unit length)

SEØ (REAL)
UNIFORM CG OFFSET
FROM EA (IN)
ENTER 1 REAL VALUE

(Enter the uniform CG offset from the elastic axis; positive - forward)

SEAØ (REAL)
UNIFORM AC OFFSET
FROM EA (+ FWD EA) (IN)
ENTER 1 REAL VALUE

(Enter the uniform area centroid offset from the elastic axis; positive - forward)

KM1Ø (REAL)
UNIFORM MASS ROG
ABOUT CHORDWISE AXIS
IN BEAMWISE DIRECTION (IN)
ENTER 1 REAL VALUE

(Enter the uniform mass radius of gyration in the beamwise [perpendicular to spanwise-chordwise plane] direction; KM1Ø $\geq$ Ø)

KM2Ø (REAL)
UNIFORM MASS ROG
ABOUT BEAMWISE AXIS
IN CHORDWISE DIRECTION (IN)
ENTER 1 REAL VALUE

(KM2Ø $\geq$ Ø)

KAØ (REAL)
UNIFORM AREA ROG
ABOUT SPANWISE AXIS
OF CROSS SECTION (IN)
ENTER 1 REAL VALUE

(Enter the uniform area radius of gyration in the chordwise-beamwise plane [KAØ = $\sqrt{Y^2 + Z^2}$]; KAØ $\geq$ Ø)

THPØ(REAL)
UNIFORM PRETWIST RATE
(DEG/IN)
ENTER 1 REAL VALUE

(Enter the uniform built-in twist per unit length)

EIYØ (REAL)
UNIFORM CHORDWISE EI
*10E - 06 INCREMENT (LBF*IN**2)
ENTER 1 REAL VALUE

(Enter the uniform change in chordwise [in-plane bending] stiffness)

EAØ (REAL)
UNIFORM SECTION EA
* 10E - 06 INCREMENT (LBF)
ENTER 1 REAL VALUE

(Enter the uniform change in spanwise elongation stiffness)

EIZØ (REAL)
UNIFORM BEAMWISE EI
*10E - 06 INCREMENT (LBF*IN**2)
ENTER 1 REAL VALUE

(Enter the uniform change in beamwise [out-of-plane bending] stiffness)

GJØ (REAL)
UNIFORM SECTION GJ
*10E - 06 INCREMENT (LBF*IN**2)
ENTER 1 REAL VALUE

(Enter the uniform change in torsional stiffness)

EB1Ø (REAL)
UNIFORM CROSS SEC
INTEGRAL INCREMENT
(IN**6)
ENTER 1 REAL VALUE

(Change in area integral of cross section with respect to local principal axes [see Volume I, Theoretical Manual])

EB2Ø (REAL)
UNIFORM CROSS SEC
INTEGRAL INCREMENT
(IN**5)
ENTER 1 REAL VALUE

EC1Ø (REAL)
UNIFORM CROSS SEC
INTEGRAL INCREMENT
(IN**6)
ENTER 1 REAL VALUE

EC1STAØ (REAL)
UNIFORM CROSS SEC
INTEGRAL INCREMENT
(IN**5)
ENTER 1 REAL VALUE

If nonuniform blade:

M (REAL)
WT PER UNIT LENGTH
INCREMENT (LB/IN)
ENTER NX REAL VALUE(S)

(Enter the change in blade weight per unit length at each blade station)

SE (REAL)
CG OFFSET FROM EA
(IN)
ENTER NX REAL VALUE(S)

(Enter the CG offset from the elastic axis at each blade station; positive - forward)

SEA (REAL)
AREA CENTROID OFFSET
FROM EA (+ FWD EA) (IN)
ENTER NX REAL VALUE(S)

(Enter the area centroid offset from the elastic axis at each blade station; positive - forward)

KM1 (REAL)
MASS ROG ABOUT LOCAL
CHORDWISE AXIS
IN BEAMWISE DIRECTION (IN)
ENTER NX REAL VALUE(S)

(Enter the mass radius of gyration in the beamwise [perpendicular to spanwise-chordwise plane] direction at each blade station; KM1(I) $\geq$ Ø)

KM2 (REAL)
MASS ROG ABOUT LOCAL
BEAMWISE AXIS
IN CHORDWISE DIRECTION (IN)
ENTER NX REAL VALUE(S)

(KM2(I) $\geq$ Ø)

KA (REAL)
AREA ROG ABOUT LOCAL
SPANWISE AXIS
OF CROSS SECTION (IN)
ENTER NX REAL VALUE(S)

(Enter the area radius of gyration in the chordwise-beamwise plane

[$KA = \sqrt{Y^2 + Z^2}$] at each blade station; KA(I) $\geq$ Ø)

THP (REAL)
PRETWIST RATE (DEG/IN)
ENTER NX REAL VALUE(S)

(Enter the built-in twist per unit length at each blade section)

EIY (REAL)
CHORDWISE EI*10E - 06
INCREMENT (LBF*IN**2)
ENTER NX REAL VALUE(S)

(Enter the change in chordwise [in-plane bending] stiffness at each blade station)

EA (REAL)
SECTION EA * 10E - 06
INCREMENT (LBF)
ENTER NX REAL VALUE(S)

(Enter the change in spanwise elongation stiffness at each blade station)

EIZ (REAL)
BEAMWISE EI*10E - 06
INCREMENT (LBF*IN**2)
ENTER NX REAL VALUE(S)

(Enter the change in beamwise [out-of-plane bending] stiffness at each blade station)

GJ (REAL)
SECTION GJ*10E - 06
INCREMENT (LBF*IN**2)
ENTER NX REAL VALUE(S)

(Enter the change in torsional stiffness at each blade station)

EB1 (REAL)
CROSS SEC INTEGRAL
INCREMENT (IN**6)
ENTER NX REAL VALUE(S)

(Change in area integral of cross section at each blade station with respect to local principal axes [see Volume I, Theoretical Manual])

EB2 (REAL)
CROSS SEC INTEGRAL
INCREMENT (IN**5)
ENTER NX REAL VALUE(S)

EC1 (REAL)
CROSS SEC INTEGRAL
INCREMENT (IN**6)
ENTER NX REAL VALUE(S)

EC1STA (REAL)
CROSS SEC INTEGRAL
INCREMENT (IN**5)
ENTER NX REAL VALUE(S)

JIL (Y OR N)
INTERNAL LOADS
ENTER 1 Y OR N VALUE

(Enter Y if internal loads [blade moments] are to be calculated from time history; see SII3)

NXIL (INTEGER)
NO. OF STATIONS
ENTER 1 INTEGER VALUE

(Enter the number of blade stations at which blade moments will be calculated; $1 \le NXIL \le NX$)

INDIL (INTEGER)
STATION INDICES
ENTER NXIL INTEGER VALUES

(Enter the indices of the NXIL blade stations in ascending order; $1 \le INDIL(I) \le NX$)

JTORIL (Y OR N)
TWIST MOMENTS
ENTER 1 Y OR N VALUE

(Enter Y if twist moments due to shear are to be calculated)

JIPIL (Y OR N)
INPLANE MOMENTS
ENTER 1 Y OR N VALUE

(Enter Y if in-plane moments are to be calculated)

JOPIL (Y OR N)
OUTPLANE MOMENTS
ENTER 1 Y OR N VALUE

(Enter Y if out-of-plane moments are to be calculated)

3.1.11 CLC2 - Linear Constraints. CLC2 is used to apply arbitrary linear constraints to the degrees of freedom of a system. Relationships between degrees of freedom are input as equations:

$$\underset{m \times n}{\begin{bmatrix} \text{CONSTANT} \\ \text{COEFFICIENT} \\ \text{MATRIX} \end{bmatrix}} \begin{Bmatrix} X_1 \\ \cdot \\ \cdot \\ \cdot \\ X_n \end{Bmatrix} = \{ 0 \} , \; m < n$$

where m is the number of constraint equations and n is the number of degrees of freedom. m implicit degrees of freedom are selected and are automatically solved for using a Gauss-Seidel algorithm yielding

$$\begin{Bmatrix} X_I \end{Bmatrix} = \underset{m \times (n-m)}{\begin{bmatrix} T \end{bmatrix}} \begin{Bmatrix} X \end{Bmatrix}$$

where X_I contains the implicit degrees of freedom and is replaced by linear combinations of the remaining degrees of freedom when the system equations are formed. X_I must be composed of existing component or system degrees of freedom; X is arbitrary.

3.1.11.1 Primary Features -

Degrees of Freedom - Up to 20 degrees of freedom. The number of constraint equations (implicit degrees of freedom) must be less than the number of degrees of freedom. An implicit degree of freedom may only be designated as such once in a given model. This includes relationships automatically formed in other technical modules. If an implicit degree of freedom is specified more than once, the last relationship in which it occurs is used.

Coupling - Rigid, pinned, and elastic coupling of components may be accomplished.

Multiple Uses - CLC2 can be used up to 20 times with a maximum of 20 degrees of freedom in a single usage and 60 implicit degrees of freedom and 1000 implicit coefficients in a model.

3.1.11.2 Input, CLC2 - The CLC2 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response

Only required input (based on previous input) is requested.

NCDF (INTEGER)
NUMBER OF DOF
ENTER 1 INTEGER VALUE(S)

(Enter the number of degrees of freedom; $\emptyset$ < NCDF $\leq$ 20)

CDFLI (DOF)
DOF NAMES
ENTER NCDF DOF NAMES (A4, 14) ONE PER LINE

(Enter the names of the degrees of freedom)

NCIDF (INTEGER)
NO. OF CONSTRAINT
EQNS
ENTER 1 INTEGER VALUE(S)

(Enter the number of constraint equations; $\emptyset$ < NCIDF < NCDF)

COEF (REAL)
COEFFICIENT MATRIX
PREPARE TO ENTER ROW/COLUMN n
ENTER NCDF/NCIDF REAL VALUES

(Enter the coefficients for the nth constraint equation/enter the coefficients for the nth degree of freedom)

NIDOF (INTEGER)
NO. OF IMPLICIT DOF
ENTER 1 INTEGER VALUE

(Ø < NIDOF ≤ NCIDF)

INDEX (INTEGER)
IMPLICIT DOF INDICES
ENTER NIDOF INTEGER VALUES

(Enter the indices of the degrees of freedom which are to be implicit in ascending order)

3.1.12 CLCØ - Linear Constraints - CLCØ is used to eliminate degrees of freedom from a system. Constraints are automatically formulated with all implicit coefficients set to zero.

3.1.12.1 Primary features -

Degrees of Freedom - Up to 40 degrees of freedom may be eliminated.

Explicit Degree of Freedom - one explicit degree of freedom (an existing degree of freedom which is not being eliminated) must be specified.

Multiple Uses - CLCØ can be used up to 20 times in a model.

3.1.12.2 Input, CLCØ - The CLCØ input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response

Only required input (based on previous input) is requested.

NCIDF (INTEGER)
NO. OF ELIMINATED DOF
ENTER 1 INTEGER VALUE

($\emptyset$ < NCIDF $\leq$ 40)

CIDFLI (DOF)
ELIMINATED DOF NAMES
ENTER NCIDF DOF NAMES (A4, I4) ONE PER LINE

CDFLI (DOF)
EXPLICIT DOF NAME
ENTER 1 DOF NAME (A4,I4)

3.1.13 CGF2 - General Force - CGF2 applies general periodic forces to the component or system degrees of freedom of a given model.

3.1.13.1 Primary Features -

Periodic Forcing - Any combination of three types of periodic forcing functions can be specified:

1. Polynomial
2. Fourier series
3. Tabular

Start and end times are input for each force application and define the period of application. The application cycle is repeated automatically, resulting in a periodic function. The periodicity may be augmented or eliminated entirely by specifying zero force over a given time period.

Polynomial Function - The user can specify a polynomial forcing function:

$$F(t) = A_0 + A_1 (t-t_1) + A_2 (t-t_1)^2 + A_3 (t-t_1)^3 + A_4 (t-t_1)^4$$

where A_n are user-input coefficients, t is the current time and t_1 is the start time of the current cycle. Separate polynomial functions can be applied to up to 40 degrees of freedom.

Fourier Series - The user can specify a Fourier series force approximation:

$$
\begin{aligned}
F(t) = A_0 &+ A_1 \cos \omega (t-t_1) + B_1 \sin \omega (t-t_1) \\
&+ A_2 \cos 2\omega (t-t_1) + B_2 \sin 2\omega (t-t_1) \\
&+ A_3 \cos 3\omega (t-t_1) + B_3 \sin 3\omega (t-t_1) \\
&+ A_4 \cos 4\omega (t-t_1) + B_4 \sin 4\omega (t-t_1)
\end{aligned}
$$

where A_n and B_n are user-input coefficients, t is the current time, t_1 is the start time of the current cycle, and ω is the frequency. Separate Fourier series force approximations can be applied to up to 40 degrees of freedom.

Tabular Function - The user can specify a tabular function with forces defined at up to 20 discreet time points. F(t) is calculated using linear interpolation. Separate force tables can be applied to up to 6 degrees of freedom.

Multiple Uses - CGF2 can be used up to 4 times in a given model.

3.1.13.2 Degrees of Freedom - All degree of freedom names must be supplied by the user.

3.1.13.3 Input, CGF2 - The CGF2 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS

response

Only required input (based on previous input) is requested.

NODF (INTEGER)
NUMBER OF DOF
ENTER 1 INTEGER VALUE

(Enter the total number of degrees of freedom to which forces will be applied; $1 < \text{NODF} \leq 40$)

NAMEDOF (DOF)
DOF NAMES
ENTER NODF DOF NAMES (A4,I4) ONE PER LINE

(Enter the names of all degrees of freedom to which forces will be applied)

IA (Y OR N)
TYPE A FORCE
(POLYNOMIAL)
ENTER 1 Y OR N VALUE

(Enter Y [yes] if polynomial forcing functions will be applied)

IB (Y OR N)
TYPE B FORCE
(FOURIER SERIES)
ENTER 1 Y OR N VALUE

IC (Y OR N)
TYPE C FORCE
(TABULAR)
ENTER 1 Y OR N VALUE

NA (INTEGER)
NO. OF TYPE A
APPLICATIONS
ENTER 1 INTEGER VALUE

(Enter the number of degrees of freedom to which type A forces will be applied; $1 \leq NA \leq NODF$)

NAMENA (DOF)
DOF NAME FOR EACH
TYPE A APPLICATION
ENTER NA DOF NAMES (A4,I4) ONE PER LINE

(Enter the names of the degrees of freedom included in NAMEDOF to which type A forces will be applied)

COEFNA (REAL)
COEFFS FOR EACH
TYPE A APPLICATION UP TO 4TH POWER
PREPARE TO ENTER ROW/COLUMN n
ENTER NA/5 REAL VALUES

(Enter the nth coefficients of the type A applications/enter the coefficients of the nth type A application)

T1NA (REAL)
START TIME FOR EACH
TYPE A APPLICATION
ENTER NA REAL VALUES

(Enter the start time [seconds] for each type A application)

T2NA (REAL)
END TIME FOR EACH
TYPE A APPLICATION
ENTER NA REAL VALUES

NB (INTEGER)
NO. OF TYPE B
APPLICATIONS
ENTER 1 INTEGER VALUE

(Enter the number of degrees of freedom to which type B forces will be applied; $1 \leq NB \leq NODF$)

NAMENB (DOF)
DOF NAME FOR EACH
TYPE B APPLICATION
ENTER NB DOF NAMES (A4,I4) ONE PER LINE

(Enter the names of the degrees of freedom included in NAMEDOF to which type B forces will be applied)

COEFNB (REAL)
COEFFS FOR EACH
TYPE B FOURIER SERIES (A_0, A_1,..., A_4, B_4)
PREPARE TO ENTER ROW/COLUMN n
ENTER NB/9 REAL VALUES

(Enter the nth coefficients of the type B applications/enter the coefficients of the nth type B application)

FQNB (REAL)
FREQ (RAD/S) FOR EACH
TYPE B APPLICATION
ENTER NB REAL VALUES

T1NB (REAL)
START TIME FOR EACH
TYPE B APPLICATION
ENTER NB REAL VALUES

(Enter the start time [seconds] for each type B application)

T2NB (REAL)
END TIME FOR EACH
TYPE B APPLICATION
ENTER NB REAL VALUES

NC (INTEGER)
NO. OF TYPE C
APPLICATIONS
ENTER 1 INTEGER VALUE

(Enter the number of degrees of freedom to which type C forces will be applied; $1 \leq NC \leq$ [NDOF, $NDOF \leq 6$] [6, $NDOF > 6$])

NAMENC (DOF)
DOF NAME FOR EACH
TYPE C APPLICATION
ENTER NC DOF NAMES (A4,I4) ONE PER LINE

(Enter the names of the degrees of freedom included in NAMEDOF to which type C forces will be applied)

NSNCn (INTEGER)
NO. OF TIME POINTS
nTH TYPE C APPLICATION
ENTER 1 INTEGER VALUE

(Enter the number of time points at which the forces for the nth type C application are to be defined; $2 \leq$ NSNCn ≤ 20)

COEFNCn (REAL)
TIMES AND FORCES
FOR nTH TYPE C APPLICATION
PREPARE TO ENTER ROW/COLUMN n
ENTER 2/NSNCn REAL VALUES

(Enter the nth time point and force/enter NSNCn time points and NSNCn forces)

3.1.14 CLG2 - Landing Gear - CLG2 is a general representation of a landing gear strut and tire. Tire and strut damping and stiffness coefficients can be nonlinear, and brakes on and off, tire scrubbing, and tire leaving the ground conditions can be modeled.

3.1.14.1 Primary Features -

Rigid Body Modes - Three translational and two rotational strut rigid body degrees of freedom are defined at the gear attachment point as shown in Figure 20.

Elastic Strut - An elastic strut elongation degree of freedom is defined along the strut principal axis.

Elastic Tire - Three implicit translational degrees of freedom are defined for the tire. The tire degrees of freedom are incorporated into the system equations as functions of the strut rigid body and elongation degrees of freedom.

Tire Scrubbing - Optionally, two additional tire translational degrees of freedom (longitudinal, lateral) are defined if ground friction and tire scrubbing are to be considered. Tire scrubbing can be evaluated for brakes on or brakes off conditions.

Nonlinear Coefficients - The stiffness and damping coefficients for the strut and tire are defined as piecewise linear functions of the displacement and velocity of the strut elongation and tire degrees of freedom. Tire damping and stiffness are zero and scrubbing degrees of freedom are inactive when the tire is not in contact with the ground (as during a time history solution).

Multiple Uses - CLG2 can be used up to 4 times in a given model.

3.1.14.2 Degrees of Freedom - The strut degree of freedom names must be supplied by the user. The tire degree of freedom names are formulated automatically as follows:

Implicit Degrees of Freedom

TIREsØ1Ø - longitudinal translation
TIREsØ2Ø - lateral translation
TIREsØ3Ø - vertical translation

Scrubbing Degrees of Freedom

FRCXsØØØ - longitudinal translation
FRCYsØØØ - lateral translation

where s = structure number (assigned during model formulation).

The positive orientations of the strut and tire degrees of freedom are presented in Figure 20.

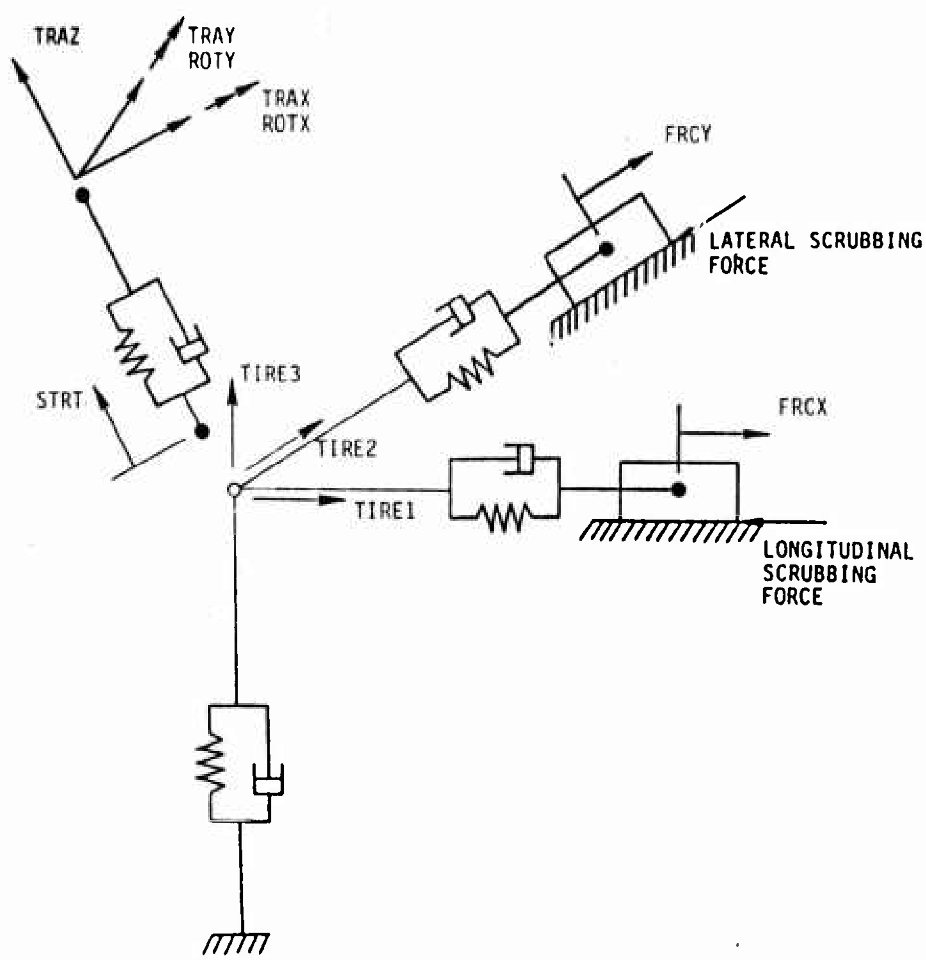


Figure 20. CLG2 Degrees of Freedom.

3.1.14.3 Input, CLG2 - The CLG2 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response

Only required input (based on previous input) is requested.

NAMEZS (DOF)
STRUT Z-TRANSLATION DOF
ENTER 1 DOF NAME (A4,I4)

(Enter the name of the strut Z-translation degree of freedom)

NAMEXS (DOF)
STRUT X-TRANSLATION
DOF
ENTER 1 DOF NAME (A4,I4)

NAMEYS (DOF)
STRUT Y-TRANSLATION
DOF
ENTER DOF NAME (A4,I4)

NAMEAX (DOF)
STRUT X-ROTATION DOF
ENTER 1 DOF NAME (A4,I4)

NAMEAY (DOF)
STRUT Y-ROTATION DOF
ENTER 1 DOF NAME (A4,I4)

NAMEDL (DOF)
STRUT ELONGATION DOF
ENTER 1 DOF NAME (A4,I4)

M1 (REAL)
TIRE MASS
ENTER 1 REAL VALUE

(Enter the tire mass [lb-sec^2/in.])

M2 (REAL)
STRUT MASS
ENTER 1 REAL VALUE

(lb-sec^2/in.)

LØ (REAL)
UNDEFORMED LENGTH OF
STRUT
ENTER 1 REAL VALUE

(in.)

ZCOS (REAL)
STRUT Z-TRAN DIR COS
WRT FUSELAGE COORDS
ENTER 3 REAL VALUE

(Enter the strut Z-translation direction cosines with respect to some fuselage coordinate system [enter the Z vector of the strut coordinate system in terms of a fuselage coordinate system in which the tire lateral, vertical, and longitudinal degrees of freedom are defined; the strut direction cosines and the undeformed strut length relate the tire and strut degrees of freedom])

XCOS (REAL)
STRUT X-TRAN DIR COS
WRT FUSELAGE COORDS
ENTER 3 REAL VALUES

(Z cross X yields the Y vector)

NKL (INTEGER)
NO. OF DEF POINTS
STRUT ELONGATION SPRING
ENTER 1 INTEGER VALUE

(Enter the number of stiffness definition points for strut elongation; $2 \leq NKL \geq 20$)

COEFKLSL (REAL)
STRUT DISPLACEMENT
ENTER NKL REAL VALUES

(Enter the displacements [in.] for which strut stiffness will be defined in ascending order)

COEFKLSR (REAL)
STRUT SPRING RATE
ENTER NKL REAL VALUES

(Enter the strut stiffnesses [lb/in.])

NCL (INTEGER)
NO. OF DEF POINTS
STRUT ELONGATION DAMPER
ENTER 1 INTEGER VALUE

(Enter the number of damping definition points for strut elongation; $2 \leq NCL \leq 20$)

COEFCLV (REAL)
STRUT VELOCITY
ENTER NCL REAL VALUES

(Enter the velocities [in./sec] for which strut damping will be defined in ascending order)

COEFCLDR (REAL)
STRUT DAMPING RATE
ENTER NCL REAL VALUES

(Enter the strut damping [lb-sec/in.])

NKX (INTEGER)
NO. OF DEF POINTS
TIRE LONGITUDINAL SPRING
ENTER 1 INTEGER VALUE

($2 \leq$ NKX ≤ 20)
(All units for TIRE same as for STRUT)

COEFKXLD (REAL)
TIRE LONG
DISPLACEMENT
ENTER NKS REAL VALUES

COEFKXSR (REAL)
TIRE SPRING RATE
ENTER NKX REAL VALUES

NCX (INTEGER)
NO. OF DEF POINTS
TIRE LONGITUDINAL DAMPER
ENTER 1 INTEGER VALUE

($2 \leq$ NCX ≤ 20)

COEFCXV (REAL)
TIRE LONG VELOCITY
ENTER NCX REAL VALUES

COEFCXDR (REAL)
TIRE DAMPING RATE
ENTER NCX REAL VALUES

NKY (INTEGER)
NO. OF DEF POINTS
TIRE LATERAL SPRING
ENTER 1 INTEGER VALUE

($2 \leq$ NKY ≤ 20)

COEFKYLD (REAL)
TIRE LAT DISPLACEMENT
ENTER NKY REAL VALUES

COEFKYSR (REAL)
TIRE SPRING RATE
ENTER NKY REAL VALUES

NCY (INTEGER)
NO. OF DEF POINTS
TIRE LATERAL DAMPER
ENTER 1 INTEGER VALUE

($2 \leq$ NCY ≤ 20)

COEFCYV (REAL)
TIRE LAT VELOCITY
ENTER NCY REAL VALUES

COEFCYDR (REAL)
TIRE DAMPING RATE
ENTER NCY REAL VALUES

NKZ (INTEGER)
NO. OF DEF POINTS
TIRE VERTICAL SPRING
ENTER 1 INTEGER VALUE

($2 \leq NKZ \leq 20$)

COEFKZVD (REAL)
TIRE VERT DISPLACEMENT
ENTER NKZ REAL VALUES

COEFKZSR (REAL)
TIRE SPRING RATE
ENTER NKZ REAL VALUES

NCZ (INTEGER)
NO. OF DEF POINTS
TIRE VERTICAL DAMPER
ENTER 1 INTEGER VALUE

($2 \leq NCZ \leq 20$)

COEFCZV (REAL)
TIRE VERT VELOCITY
ENTER NCZ REAL VALUES

COEFCZDR (REAL)
TIRE DAMPING RATE
ENTER NCZ REAL VALUES

FRIC (Y OR N)
GROUND FRICTION
ENTER 1 Y OR N VALUE

(Enter Y if ground friction is to be considered, else N)

BRAKE (Y OR N)
BRAKES ON
ENTER 1 Y OR N VALUE

(Enter Y if brakes on condition is to be in effect, else N)

SCOX (REAL)
LONG SCRUBBING COEFF
ENTER 1 REAL VALUE

(Enter dimensionless coefficient of friction - longitudinal direction; SCOX $\geq$ Ø)

SCOY (REAL)
LAT SCRUBBING COEFF
ENTER 1 REAL VALUE

3.1.15 CLS2 - Lifting Surface (Modal). CLS2 is a specialized modal representation for a planar lifting surface structure with a rigid hinged control surface.

3.1.15.1 Primary Features -

Rigid Body Modes - Up to three translational and three rotational rigid body degrees of freedom defined at the wing attachment point as shown in Figure 21.

Elastic Modes - Up to 10 normal modes can be defined at up to 20 stations along the principal (spanwise) axis of the structure. Each mode can include in-plane and out-of-plane displacements and slopes and torsional displacements.

Implicit Degree of Freedom - Up to three translational and three rotational implicit degrees of freedom can be defined at up to 20 node points in the plane of the structure. These implicit degrees of freedom are arbitrary and can be degrees of freedom of other components, which will result in the automatic coupling of the components. Implicit degrees of freedom are automatically processed as functions of the rigid body and elastic degrees of freedom of this component when the system equations are formed.

Hinged Control Surface - A control surface angular degree of freedom can optionally be defined. Two reaction points define the hinge line.

Symmetry Option - The user can mirror the degrees of freedom and create a structure which is symmetric about its centerline.

Aerodynamic Forcing - Optional use of FLA2 for steady lifting surface aerodynamic forces (see paragraph 3.2 for details).

Multiple Uses - CLS2 can be used up to 4 times in a given model.

3.1.15.2 Degrees of Freedom - Degree of freedom names are formulated automatically as follows:

Rigid Body Degrees of Freedom

XATTsØØr - translation in spanwise direction
YATTsØØr - translation in chordwise direction
ZATTsØØr - out-of-plane translation
PATTsØØr - pitch about spanwise axis
FATTsØØr - flapping about chordwise axis (roll)
SATTsØØr - sweeping about out-of-plane axis (yaw)

Control Surface Degrees of Freedom

DELTsØØr - rotation about hinge axis

Elastic Degrees of Freedom

QWNGsmØr

Implicit Degrees of Freedom

XAUXsØnr - local x-translation
YAUXsØnr - local y-translation
ZAUXsØnr - local z-translation
PAUXsØnr - rotation about local x-axis
FAUXsØnr - rotation about local y-axis
SAUXsØnr - rotation about local z-axis

where s = structure number (assigned during model formulation)
m = elastic mode number
n = node number
r = coordinate system

The user has the option of choosing a left or right-handed coordinate system. In addition, if the symmetric structure option is exercised, left and right-handed coordinate systems will be defined for the two respective structures so that degree of freedom definitions will be consistent.

The positive orientations of the rigid body, control surface, and implicit degrees of freedom are presented in Figure 21. The modal displacements at each station are defined with respect to the wing coordinate system, and the control surface and implicit degrees of freedom are defined locally.

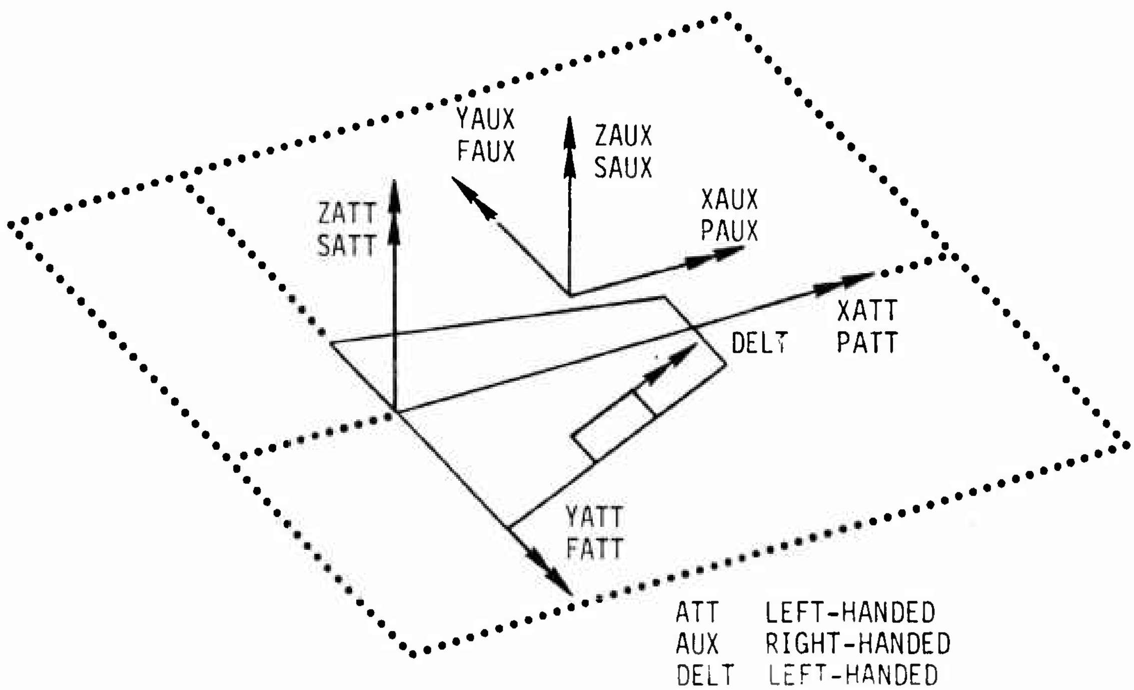


Figure 21. CLS2 Degrees of Freedom.

3.1.15.3 <u>Input, CLS2</u>. The CLS2 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS

response

Only required input (based on previous input) is requested.

WING (ALPHA)
PORT OR STARBOARD
ENTER PORT OR STAR
PORT ASSUMED IF OTHER THAN PORT OR STAR IS ENTERED

(Enter PORT if a right-handed coordinate system is elected, else STAR)

RBM (Y OR N)
RIGID BODY MODES
ENTER 1 Y OR N VALUE

(Enter Y [yes] if rigid body modes are to be included, else N [no])

ISTA (Y OR N)
STATIONWISE (X)
ENTER 1 Y OR N VALUE

(Y if stationwise [spanwise] degree of freedom is elected, else N)

ICHORD (Y OR N)
CHORDWISE (Y)
ENTER 1 Y OR N VALUE

IOP (Y OR N)
OUT OF PLANE (Z)
ENTER 1 Y OR N VALUE

IPITCH (Y OR N)
PITCHING (X--X)
ENTER 1 Y OR N VALUE

IFLAP (Y OR N)
FLAPPING (Y--Y)
ENTER 1 Y OR N VALUE

ISWEEP (Y OR N)
SWEEPING (Z--Z)
ENTER 1 Y OR N VALUE

ATTXY (REAL)
WING ATTACH NODE XY
X Y COORDINATES OF ROOT OF WING ELASTIC AXIS (IN)
RIGID BODY DOF DEFINED HERE
ENTER 2 REAL VALUES

(Enter the coordinates of the wing attachment point [inches]; the origin of the wing coordinate system is defined by the user)

AILERON (Y OR N)
CONTROL SURF OPTION
ENTER 1 Y OR N VALUE

AILCG (REAL)
AILERON CG XY (IN)
ENTER 2 REAL VALUES

(Enter the coordinates of the aileron CG)

AILCI (Y OR N)
REFERENCE AXES AT CG
YES = INERTIAS ABOUT AILERON CG
NO = INERTIAS ABOUT AILERON HINGE
ENTER 1 Y OR N VALUE

AILINERT (REAL)
AILERON INERTIAS
1. WEIGHT (LB)
2. IXX (LB-IN**2) MOI ABOUT HINGE/AXIS THROUGH CG
3. IYY MOI ABOUT AXIS THROUGH CG
4. IZZ MOI ABOUT OUT-OF-PLANE AXIS AT HINGE/CG
5. IYZ YZ PRODUCT OF INERTIA AT HINGE/CG
ENTER 5 REAL VALUES

HINGE (REAL)
REACTION POINTS
ROW 1 -- INBOARD XY
ROW 2 -- OUTBOARD XY
PREPARE TO ENTER ROW/COLUMN N
ENTER 2 REAL VALUES

WING CG (REAL)
WING CG XY (IN)
ENTER 2 REAL VALUES

(Enter the coordinates of the wing CG)

WING CI (Y OR N)
REFERENCE AXES AT CG
YES = INERTIAS ABOUT WING CG
NO = INERTIAS ABOUT WING ATTACH NODE

WINGINER (REAL)
WING INERTIAS
1. WEIGHT (LB)
2. IXX
3. IYY
4. IZZ
5. IXY
6. IYZ (LB-IN**2)
ENTER 6 REAL VALUES

NMODE (INTEGER)
NO. OF ELASTIC MODES
ENTER 1 INTEGER VALUE

($\emptyset$ = NMODE $\leq$ 10)

NSTA (INTEGER)
NO. OF MODAL STATIONS
ENTER 1 INTEGER VALUE

(1 $\leq$ NSTA $\leq$ 20)

MSTAT (REAL)
MODE SHAPE STATIONS
ENTER NSTA REAL VALUES

(Enter the mode shape stations [inches] in ascending order)

IPCOMP (Y OR N)
IP COMPONENTS
ENTER 1 Y OR N VALUE

(Enter Y if there are any in-plane modal components)

OPCOMP (Y OR N)
OP COMPONENTS
ENTER 1 Y OR N VALUE

(Enter Y if there are any out-of-plane modal components)

TORCOMP (Y OR N)
TORSION COMPONENTS
ENTER 1 Y OR N VALUE

IPMODE (REAL)
MODES IP COMPONENT
NMODE ROWS, NSTA COLUMNS
PREPARE TO ENTER ROW/COLUMN n
ENTER NSTA/NMODE REAL VALUES

(Enter the in-plane component of the nth mode shape at the NSTA stations/enter the in-plane components of the NMODE mode shapes at the nth station)

IPSLOP (REAL)
IP SLOPES
PREPARE TO ENTER ROW/COLUMN n
ENTER NSTA/NMODE REAL VALUES

OPMODE (REAL)
MODES OP COMPONENT
PREPARE TO ENTER ROW/COLUMN n
ENTER NSTA/NMODE REAL VALUES

OPSLOP (REAL)
OP SLOPES
PREPARE TO ENTER ROW/COLUMN n
ENTER NSTA/NMODE REAL VALUES

TORMODE (REAL)
MODES TOR COMPONENT
PREPARE TO ENTER ROW/COLUMN n
ENTER NSTA/NMODE REAL VALUES

MODEINERT (REAL)
MODAL PROPERTIES
COLUMN 1. MODAL MASS -- 2.DAMPING -- 3. FREQUENCY
NMODE ROWS, 3 COLUMNS
PREPARE TO ENTER ROW/COLUMN n
ENTER 3/NMODE REAL VALUES

(Enter the nth modal mass [slugs], modal damping [percent of critical damping], and modal frequency [Hz]/enter NMODE modal masses, modal dampings, and modal frequencies)

NAUX (INTEGER)
NO. OF AUX NODES
OPTIONAL AUXILIARY NODES AT WHICH IMPLICIT DOFS ARE DEFINED
THEY NEED NOT BEAR ANY RELATION TO MODE STATIONS
ENTER 1 INTEGER VALUE ($\emptyset \leq$ NAUX $\leq$ 20)

XYAUX (REAL)
XY FOR EACH AUX NODE
PREPARE TO ENTER ROW/COLUMN n
ENTER 2/NAUX REAL VALUES

(Enter the auxiliary node coordinates)

NODOF (Y OR N)
DOF Y OR N FOR NODES
COLUMNS--Y OR N FOR XAUX YAUX ZAUX PAUX FAUX SAUX;
6 COLUMNS, NAUX ROWS
PREPARE TO ENTER ROW n
ENTER 6 Y OR N VALUES (35A2)

LOCVEC (REAL)
LOCAL XY VECTORS
DIRECTION COSINES FOR LOCAL XY VECTORS
WRT WING COORDINATE SYSTEM
6 COLUMNS, NAUX ROWS
PREPARE TO ENTER ROW/COLUMN n
ENTER 6/NAUX REAL VALUES

(Enter the X and Y vectors of the local coordinate system at each auxiliary node in terms of the wing coordinate system [X cross Y yields the Z vector]; see Figure 19)

```
YZSYMM (Y OR N)
SYMMETRY OPTION
OPTION TO CREATE IDENTICAL WING CHIRALLY REFLECTED
THROUGH YZ SYMMETRY PLANE
ENTER 1 Y OR N VALUE
```

3.1.16 CGLØ - Global Transformation. CGLØ is model-dependent and forms the matrices for transformation of gravitational or centrifugal accelerations from the global coordinate system to the individual component coordinate systems (see paragraph 3.4).

3.1.16.1 Primary Features -

Component Sequence Numbers - The user specifies the sequence numbers of the data set/data members in the model for which transformation is required.

Direction Cosines - The component X and Y coordinate axes expressed as vectors in terms of the global coordinates are requested.

Use - One use per model.

3.1.16.2 Degrees of Freedom - One arbitrary degree of freedom must be specified.

3.1.16.3 Input, CGLØ - The CGLØ input prompts are shown below in capital letters. Input prompt format is:

```
VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
```

response

Only required input (based on previous input) is requested.

```
NCGL (INTEGER)
NO. OF COMPONENTS
FOR WHICH GLOBAL TRANSFORM WILL BE PERFORMED
ENTER 1 INTEGER VALUE
```

($1 \leq NCGL \leq 20$)

IGSEQ (INTEGER)
SEQUENCE NUMBERS
OF COMPONENTS SELECTED
ENTER NCGL INTEGER VALUES

(Enter the sequence numbers in ascending order)

XYDG (REAL)
COMPONENT X, Y VECTORS
IN TERMS OF GLOBAL SYSTEM
PREPARE TO ENTER ROW/COLUMN n
ENTER NCGL/6 REAL VALUES

(Enter the X and Y vectors of the component coordinate system in terms of the global coordinate system [X cross Y yields the Z vector]; see Figure 19)

CDFLI (DOF)
DOF NAME
ENTER 1 DOF NAME (A4,I4)

3.1.17 CSF3 - Nonlinear Spring, Damper System. CSF3 represents an arbitrary, constant coefficient set of second-order differential equations of the form

$$M\ddot{X} + C\dot{X} + KX + C_2\dot{X}^2 + C_3\dot{X}^3 + K_2X^2 + K_3X^3 + DX\dot{X} + AX\dot{X}^2 + BX^2\dot{X} = F$$

where M, C, K, C_2, C_3, K_2, K_3, D, A, and B are coefficient matrices, F is a vector, and X is the displacement vector of the system degrees of freedom.

3.1.17.1 Primary Functions -

Constant Coefficients - Coefficients may be input for up to 40 degrees of freedom.

Coupling - Degrees of freedom may be degrees of freedom of other components, which will result in the automatic coupling of the components. Through this mechanism, a CSF3 data set may be used to modify the constant coefficients of the system equations of a model.

Global Reference - In conjunction with the global reference option (see paragraph 3.4), the user may specify transformations to the local degree of freedom coordinate systems of lumped parameter systems.

Multiple Uses - CSF3 can be used up to 20 times in a given model.

3.1.17.2 <u>Degrees of Freedom</u> - All degree of freedom names must be supplied by the user.

3.1.17.3 <u>Input, CSF3</u> - The CSF3 input prompts are shown below in capital letters. Input prompt format is:

```
VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response
```

Only required input (based on previous input) is requested.

```
NCDF (INTEGER)
NO. OF DOF
ENTER 1 INTEGER VALUE
```

(Enter the number of degrees of freedom; $1 \leq \text{NCDF} \leq 40$)

```
SDFLI (DOF)
DOF NAMES
ENTER NCDF DOF NAMES (A4,I4) ONE PER LINE

CM (REAL)
M
TYPE MATRIX
(Ø = NULL), (1 = DIAGONAL), (2 = SYMMETRIC), (3 = GENERAL)

if 1:

INPUT NCDF DIAGONAL REAL VALUES

if 2:

PREPARE TO ENTER ROW n
ENTER n REAL VALUES

if 3:

PREPARE TO ENTER ROW/COLUMN n
ENTER NCDF REAL VALUES

CC (REAL)
C
.
.
.
PREPARE TO ENTER ROW/COLUMN n
ENTER NCDF REAL VALUES
```

```
CK (REAL)
K
.
.
.
PREPARE TO ENTER ROW/COLUMN n
ENTER NCDF REAL VALUES

CC2
C2
.
.
.
PREPARE TO ENTER ROW/COLUMN n
ENTER NCDF REAL VALUES

CC3
C3
.
.
.
PREPARE TO ENTER ROW/COLUMN n
ENTER NCDF REAL VALUES

CK2
K2
.
.
.
PREPARE TO ENTER ROW/COLUMN n
ENTER NCDF REAL VALUES

CK3
K3
.
.
.
PREPARE TO ENTER ROW/COLUMN n
ENTER NCDF REAL VALUES

CA
A
.
.
.
ENTER NCDF REAL VALUES
PREPARE TO ENTER ROW/COLUMN n
```

CB
B
.
.
.
PREPARE TO ENTER ROW/COLUMN n
ENTER NCDF REAL VALUES

CD
D
.
.
.
PREPARE TO ENTER ROW/COLUMN n
ENTER NCDF REAL VALUES

CF
F
ENTER NCDF REAL VALUES

IGR (Y OR N)
GLOBAL REFERENCE
ENTER 1 Y OR N VALUE

(Enter Y [yes] if global reference is to be used, else N[no])

LDC (REAL)
LOCAL DOF VECTORS
IN TERMS OF COMPONENT REFERENCE COORDINATE SYSTEM
PREPARE TO ENTER ROW/COLUMN n
ENTER 3/NCDF REAL VALUES

(Enter the local degree of freedom vectors in terms of the component coordinate system)

3.2 FORCE TECHNOLOGY MODULES

A force module represents an algorithm for computing the applied forces for a given formulation of a set of second-order differential equations. Forces are incorporated into associated component equations of motion and, in general, are of the form

$$F(t,x,\dot{x}) + F_0$$

where x represents the component degrees of freedom and F_0 is some constant force vector.

It follows that a given force module may only be used in conjunction with specific component modules.

3.2.1 FSS1 - Sinusoidal Shaker. FSS1 simulates a sinusoidal applied force at any CSF1 degree of freedom. The forcing function is of the form

$$A \cos \omega t + B \sin \omega t$$

where ω is the shaker frequency and A and B are the amplitude components.

3.2.1.1 Primary Features -

Multiple Uses - FSS1 may be used up to 20 times in a given model.

3.2.1.2 Input, FSS1 - The FSS1 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS (FORTRAN READ FORMAT)
response

Only required input (based on previous input) is requested.

DOF (DOF)
DOF FORCED BY SHAKER
ENTER 1 DOF NAME (A4, I4)

(The name supplied must be a degree of freedom of the CSF1 component with which FSS1 is used; if it is not, no force will be applied and no warning will be given)

FREQ (REAL)
FREQUENCY (HZ)
ENTER 1 REAL VALUE

(Enter the shaker frequency)

COSC (REAL)
COSINE COMPONENT
ENTER 1 REAL VALUE

(Enter A; force [lb]; moment [lb-in.]; note that CSF1 degrees of freedom are general and may represent displacements or rotations)

SINC (REAL)
SINE COMPONENT
ENTER 1 REAL VALUE

(Enter B)

3.2.2 FRAØ - Rotor Aerodynamics, Linear (2-d). FRAØ generates blade lift, drag, and pitching moment distributions from a constant lift curve slope with cutoff at the stall angle and constant drag and pitching moment coefficients.

3.2.2.1 Primary Features -

Inflow Distribution - Uniform inflow or Glauert induced flow distribution may be elected.

Aerodynamic Coefficient Factors - Arbitrary factors may be applied to C_L, C_D, C_M versus nondimensional rotor radius in order to simulate blade cutouts, tip loss effects, or other losses.

3.2.2.2 Input, FRAØ - Blade data and the system state vector are passed automatically at solution time. Air density ratio and speed of sound are input as global variables during model formulation. The FRAØ input prompts are shown below in capital letters. Input prompt format is

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS

response

Only required input (based on previous input) is requested.

VAIR (REAL)
WIND VELOCITY
VX, VY, VZ (FT/SEC) IN HUB SYSTEM
ENTER 3 REAL VALUES

(Enter the relative wind velocity vector with respect to the rotor hub coordinate system)

ALAMDA (REAL)
AVG INDUCED VELOCITY
POS UP HUB REF
ENTER 1 REAL VALUE

(Enter nondimensional uniform induced velocity; positive = +ZHUB)

ILINEA (Y OR N)
INDUCED VLCTY VARIES
ENTER 1 Y OR N VALUE

(Enter Y if Glauert induced flow distribution is elected)

CONST (REAL)
CONST G: GLAUERT EQN
ALAMDA * (1 + G * (X/R) * COS (PSI))
ENTER 1 REAL VALUE

(Enter Glauert constant, G, of Glauert induced flow distribution, λ_0 [1 + G (x/r) cos ψ], where λ_0 (ALAMDA) is the uniform induced velocity, x/r is the nondimensional radial distance, and ψ is the azimuth angle; $0 \leq G \leq 1$)

XAC (REAL)
A/C OFFSET FROM F A
+ FWD (PERCENT OF CHORD)
ENTER 1 REAL VALUE

(Enter the aerodynamic center offset from the feathering axis; percent offset is constant along the blade span)

AØ (REAL)
LIFT CURVE SLOPE
(1/DEG)
ENTER 1 REAL VALUE

ALOL (REAL)
ALPHA FOR ZERO LIFT
(DEG)
ENTER 1 REAL VALUE

CDØ (REAL)
DRAG COEFFICIENT
ENTER 1 REAL VALUE

CMØ (REAL)
MOMENT COEFFICIENT
ENTER 1 REAL VALUE

ASTALL (REAL)
POS STALL ANGLE
(DEG)
ENTER 1 REAL VALUE

(Enter positive stall angle; for $\alpha \geq \alpha_{stall}$, C_L is constant)

NXA (INTEGER)
NO. OF AERO
STATIONS
ENTER 1 INTEGER VALUE(S)

(Enter the number of blade stations at which the aerodynamic coefficients are to be defined; $5 \leq NXA \leq 40$)

XSTA (REAL)
NON-DIMENSL STATIONS
ENTER NXA REAL VALUE(S)

(Enter the nondimensional blade stations in ascending order; need not be coincident with component module blade stations; $\emptyset \leq XSTA(I) \leq 1$)

CHORD (REAL)
CHORD FOR STATIONS
ENTER NXA REAL VALUE(S)

(Enter the blade chord [in.] for each station)

NY (INTEGER)
NO. OF FACTOR
STATIONS
ENTER 1 INTEGER VALUE(S)

(Enter the number of blade stations at which aerodynamic coefficient factors are to be applied; $2 \leq NY \leq 20$)

XF (REAL)
NON-DIMENSL STATIONS
ENTER NY REAL VALUE(S)

(Enter the factor stations in ascending order; need not be coincident with component module blade stations or aerodynamic coefficient blade stations; must start with Ø and end with 1; $\emptyset \leq XF(I) \leq 1$)

FL (REAL)
LIFT COEFFICIENT
FACTORS
ENTER NY REAL VALUE(S)

(Enter the factor to be applied to C_L at each factor station)

FD (REAL)
DRAG COEFFICIENT
FACTORS
ENTER NY REAL VALUE(S)

FM (REAL)
PITCHING MOMENT
COEFFICIENT FACTORS
ENTER NY REAL VALUE(S)

3.2.3 FRA2 - Rotor Aerodynamics, Tabular (2-d). FRA2 generates blade lift, drag, and pitching moment distributions from user-generated airfoil tables.

3.2.3.1 Primary Features -

Airfoil Tables - Up to five airfoil tables may be accessed for a given usage of FRA2. A table consists of C_L, C_D, and C_M given as functions of angle of attack and Mach number. A table is addressed by name and may reside on a user data file or be temporarily created on the run data file (see paragraphs 2.1, File Assignments, and 2.2.3, CRE, for details of the creation and use of airfoil tables).

Inflow Distribution - Uniform inflow, Glauert induced flow distribution, or induced velocity map may be elected.

The induced velocity map is treated as temporary data and is read directly from a sequential file.

The FORTRAN statements used to create a table in the proper format are as follows:

```
      REWIND NFILE
      WRITE (NFILE,1ØØ) NX1,NOPSI,(XIN(J), J=1,NX1)
      DO 1Ø I=1, NOPSI
      WRITE (NFILE,2ØØ) BPSI (I)
1Ø    WRITE (NFILE,2ØØ) (ALAMØ(I,J),J=1,NX1)
1ØØ   FORMAT (2I1Ø/(8F1Ø.6))
2ØØ   FORMAT (8F1Ø.6)
```

The variables are defined as

NFILE	-	Logical unit number of sequential file
NX1	-	Number of blade stations (Ø ≤ NX1 ≤ 2Ø)
NOPSI	-	Number of azimuth stations (Ø ≤ NOPSI ≤ 36)
BPSI(I)	-	Ith azimuth station at which induced velocity is available in table, deg (0-360)

XIN(J) - Jth radial station at which induced velocity is available in table, in.

ALAMØ(I,J)- Induced velocity corresponding to the Ith azimuth station and the Jth radial station, nondimensional, positive up, shaft reference

The radial stations need not coincide with the blade stations used in the component module.

Aerodynamic Coefficient Factors - Arbitrary factors may be applied to C_L, C_D, C_M versus nondimensional rotor radius in order to simulate blade cutouts, tip loss effects, or other losses.

3.2.3.2 Input, FRA2 - Blade data and the system state vector are passed automatically at solution time. Air density ratio and speed of sound are input as global variables during model formulation. The FRA2 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response

Only required input (based on previous input) is requested.

VAIR (REAL)
WIND VELOCITY
VX, VY, VZ W.R.T HUB SYS (FT/SEC)
ENTER 3 REAL VALUES

(Enter the relative wind velocity vector with respect to the rotor hub coordinate system)

ALAMDA (REAL)
NONDIM INDUCED VEL
+ UPWARD (AVERAGE)
ENTER 1 REAL VALUE

(Enter nondimensional uniform induced velocity; positive ≃ +ZHUB)

INDUC (INTEGER)
INDUCED VEL TYPE
ENTER 1 INTEGER VALUE

(1. Uniform inflow
2. Glauert induced flow distribution
3. Induced velocity map)

CONST (REAL)
GLAUERT CONSTANT
ENTER 1 REAL VALUE

(Enter Glauert constant, G, of Glauert induced flow distribution, λ_0 [1 + G (x/r) cos ψ], where λ_0 (ALAMDA) is the uniform induced velocity, x/r is the nondimensional radial station, and ψ is the azimuth angle; $\emptyset \leq G \leq 1$)

SFDES (SEQUENTIAL FILE)
SEQ FILE DESCRIPTION
ENTER SEQUENTIAL FILE DESCRIPTION (20 CHARACTERS)

NXA (INTEGER)
NO. OF AERO STAS
ENTER 1 INTEGER VALUE(S)

(Enter the number of blade stations at which the aerodynamic coefficients are to be defined; $5 \leq NXA \leq 40$)

SAERO (REAL)
NONDIM AERO STAS
ENTER NXA REAL VALUE(S)

(Enter the nondimensional blade [radial] stations in ascending order; need not be coincident with component module blade stations; $\emptyset \leq SAERO(I) \leq 1$)

NUMAF (INTEGER)
NO. AIRFOIL TABLES
ENTER 1 INTEGER VALUE(S)

($1 \leq NUMAF \leq 5$)

AFTABn (AIRFOIL DS)
NAME AF TABLE n
ENTER AIRFOIL DATASET NAME

NUMAFn (INTEGER)
NO. STAS AF n
ENTER 1 INTEGER VALUE(S)

(Enter the number of stations with which table n will be used; $1 \leq NUMAFn \leq NXA$)

STA-AFn (INTEGER)
STAS AF n
ENTER NUMAFn INTEGER VALUE(S)

(Enter the indices of the stations SAERO(I) with which table n will be used)

XAC (REAL)
A/C OFFSET FROM F A
+ FWD (PERCENT OF CHORD)
ENTER 1 REAL VALUE

(Enter the aerodynamic center offset from the feathering axis; percent offset is constant along the blade span)

CHORD (REAL)
CHORD (IN)
ENTER NXA REAL VALUE(S)

(Enter the blade chord at each station; CHORD $\geq$ Ø)

NX (INTEGER)
NO. AERO FACTOR STAS
ENTER 1 INTEGER VALUE(S)

(Enter the number of stations at which aerodynamic coefficient factors are to be applied; $2 \leq NX \leq 20$)

XF (REAL)
NONDIM FACTOR STAS
ENTER NX REAL VALUE(S)

(Enter the factor stations in ascending order; need not be coincident with component module blade stations or aerodynamic coefficient blade stations; must start with 0 and end with 1; Ø $\leq$ XF(I) $\leq$ 1)

FL (REAL)
FACTORS FOR CL
ENTER NX REAL VALUE(S)

(Enter the factor to be applied to C_L at each factor station)

FD (REAL)
FACTORS FOR CD
ENTER NX REAL VALUE(S)

FM (REAL)
FACTORS FOR CM
ENTER NX REAL VALUE(S)

3.2.4 FRA3 - Rotor Aerodynamics, General. FRA3 is a general algorithm for the calculation of blade lift, drag, and pitching moment distributions. Steady and unsteady aerodynamics may be utilized, and the steady state lift,

drag, and pitching moment coefficients can be calculated automatically or be obtained from user-generated airfoil tables (see Volume I, Theoretical Manual).

3.2.4.1 Primary Features -

Airfoil Tables - Up to five airfoil tables may be accessed for a given usage of FRA3. A table consists of C_L, C_D, and C_M given as functions of angle of attack and Mach number. A table is addressed by name and may reside on a user data file or be temporarily created on the run data file (see paragraphs 2.1, File Assignments, and 2.2.3, CRE, for details of the creation and use of airfoil tables).

Automatic C_L, C_D, C_M Calculations - See Volume I, Theoretical Manual.

Unsteady Aerodynamics - See Volume I, Theoretical Manual.

Inflow Distribution - The inflow distribution can be calculated automatically or be read in from a table (see Volume I, Theoretical Manual).

The induced velocity map is treated as temporary data and is read directly from a sequential file.

The FORTRAN statements used to create a table in the proper format are as follows:

```
REWIND NFILE
WRITE (NFILE,1ØØ)(RWKTTL(I), I=1, 8),NMUR,NLMR,NHHR,NRSR
IF (NMUR.GT.1) WRITE (NFILE,2ØØ) (WKMUR(I), I=1, NMUR)
IF (NLMR.GT.1) WRITE (NFILE,2ØØ) (WKLMR(I), I=1, NLMR)
IF (NRSR.GT.Ø) WRITE (NFILE,2ØØ) (XR(I), I=NRSR)
NHM = 1 + 2 * NHHR
```

```
      DO 2Ø I=1, NMUR
      DO 2Ø J=1, NLMR
      DO 2Ø K=1, NRSR
      WRITE (NFILE,2ØØ) (WKRTR(L,K,J,I), L=1, 7)
      LMT = 7
   1Ø CONTINUE
      IF (NHM.LE.LMT) GO TO 2Ø
      LL = LMT + 1
      LMT= LMT + 6
      WRITE (NFILE,3ØØ) (WKRTR(L,K,J,I), L=LL,LMT)
      GO TO 1Ø
   2Ø CONTINUE
  1ØØ FORMAT (8A4,8X,4I3)
  2ØØ FORMAT (7F1Ø.Ø)
  3ØØ FORMAT (1ØX,6F1Ø.Ø)
```

The variables are defined as

NFILE	- Logical unit number of sequential file
RWKTTL(I)	- 32-character description
NMUR	- Number of advance ratios ($1 \leq$ NMUR ≤ 3)
NLMR	- Number of inflow ratios ($1 \leq$ NLMR ≤ 2)
NHHR	- Order of highest harmonic ($Ø \leq$ NHHR ≤ 6)
WKMUR(I)	- Advance ratios
WKLMR(I)	- Inflow ratios
XR(I)	- Nondimensional radial stations
WKRTR(L,K,J,I)	- Harmonic contents table

The radial stations need not coincide with the blade stations used in the component module.

<u>Aerodynamic Coefficient Factors</u> - Arbitrary factors may be applied to C_L, C_D, C_M versus nondimensional rotor radius in order to simulate blade cutouts, tip loss effects, or other losses.

3.2.4.2 <u>Input, FRA3</u> - Blade data and the system state vector are passed automatically at solution time. Air density ratio and speed of sound are input as global variables during model formulation. The FRA3 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response

Only required input (based on previous input) is requested.

IEQS (Y OR N)
AERODYNAMICS BY EQS
ENTER 1 Y OR N VALUE

(Enter Y if steady state C_L, C_D, and C_M are to be calculated automatically, else N, read from airfoil tables)

INFTAB (Y OR N)
INDUCED VEL P: TABLE
ENTER 1 Y OR N VALUE

(Enter Y if induced velocity map is to be used, else N, inflow distribution calculated automatically)

SFDES (SEQUENTIAL FILE)
SEQ FILE DESCRIPTION
ENTER SEQUENTIAL FILE DESCRIPTION (20 CHARACTERS)

IUNSTD (Y OR N)
UNSTEADY AERODYNAMICS
ENTER 1 Y OR N VALUE

(Enter Y if unsteady aerodynamics, else N [see Volume I, Theoretical Manual])

VAIRH (REAL)
WIND VELOCITY
VX, VY, VZ W.R.T HUB SYS (FT/SEC)
ENTER 3 REAL VALUES

(Enter the relative wind velocity vector with respect to the rotor hub coordinate system)

ASTALL (REAL)
STALL ANGLE (DEG)
ENTER 1 REAL VALUE

(Enter the blade stall angle)

RFCT (REAL)
INDUCED VEL FACTOR
ENTER 1 REAL VALUE

(May be used to adjust the calculated average induced velocity [see Volume I, Theoretical Manual])

TIPLOC (REAL)
TIP LOSS COEFFICIENT
ENTER 1 REAL VALUE

(May be applied to account for the change in the thrust coefficient [induced velocity calculation] due to blade tip vorticity [see Volume I, Theoretical Manual]; Ø ≤ TIPLOC ≤ 1)

XH (REAL)
HUB EXTENT (IN)
ENTER 1 REAL VALUE

(May be applied to account for the change in the thrust coefficient [induced velocity calculation] due to the presence of the hub [see Volume I, Theoretical Manual]; XH ≥ Ø)

ALT (REAL)
VEHICLE ALTITUDE (FT)
ENTER 1 REAL VALUE

(To determine ground effect [induced velocity calculation, see Volume I, Theoretical Manual]; ALT ≥ Ø)

K27 (REAL)
TIP VORTEX
COEFFICIENT
ENTER 1 REAL VALUE

(To modify inflow distribution [as a function of blade tip vorticity] with respect to blade station and azimuth [see Volume I, Theoretical Manual]; K27 ≥ Ø)

CDØ (REAL)
BLADE DRAG COEFF
AT ALFA = Ø, M = .3
ENTER 1 REAL VALUE

(Enter the blade drag coefficient for zero angle of attack at .3 Mach number, steady C_L, C_D, C_M calculations; CDØ ≥ Ø)

Q1C (REAL)
Q1C COEFFICIENT
MEFF = M * (COS Q1C * GAMA) ** Q2C
ENTER 1 REAL VALUE

(Empirical coefficient in effective Mach number equation, steady C_L, C_D, C_M calculations [see Volume I, Theoretical Manual]; Q1C ≥ Ø)

Q2C (REAL)
Q2C COEFFICIENT
MEFF = M * (COS Q1C * GAMA) ** Q2C
ENTER 1 REAL VALUE

(See above)

ALAMDA (REAL)
NONDIM INDUCED
VELOCITY
POSITIVE UPWARD
ENTER 1 REAL VALUE

(Enter nondimensional uniform induced velocity; positive ≃ +ZHUB)

NXA (INTEGER)
NO. OF STATIONS
ENTER 1 INTEGER VALUE(S)

(Enter the number of blade stations at which the aerodynamic coefficients are to be defined; $5 \leq$ NXA ≤ 40)

XAERO (REAL)
NONDIM AERO STATIONS
ENTER NXA REAL VALUE(S)

(Enter the nondimensional blade [radial] stations in ascending order; need not be coincident with component module blade stations; $0 \leq$ XAERO(I) ≤ 1)

NUMAF (INTEGER)
NO. AIRFOIL TABLES
ENTER 1 INTEGER VALUE(S)

($1 \leq$ NUMAF ≤ 5)

AFTABn (AIRFOIL DS)
NAME AF TABLE n
ENTER AIRFOIL DATASET NAME

NUMAFn (INTEGER)
NO. STAS AF n
ENTER 1 INTEGER VALUE(S)

(Enter the number of stations with which table n will be used; $1 \leq$ NUMAFn $\leq$ NXA)

STA-AFn (INTEGER)
STAS AF n
ENTER NUMAFn INTEGER VALUE(S)

(Enter the indices of the stations XERO(I) with which table n will be used)

XACC (REAL)
A/C OFFSET FROM E C
+ FWD (PERCENT OF CHORD)
ENTER NXA REAL VALUE(S)

(Enter the aerodynamic center offset from the blade elastic center at each station)

CHORDC (REAL)
CHORD (IN)
ENTER NXA REAL VALUE(S)

(Enter the blade chord at each station; CHORD(I) $\geq$ Ø)

RAERO (REAL)
DATA FOR EQS
SEE MANUAL FOR DETAILS
ENTER 35 REAL VALUES

(If steady state C_L, C_D, C_M are to be calculated automatically, input parameters [see Volume I, Theoretical Manual])

NX (INTEGER)
NO. AERO FACTOR
STATIONS
ENTER 1 INTEGER VALUE(S)

(Enter the number of stations at which aerodynamic coefficient factors are to be applied; 2 $\leq$ NX $\leq$ 20)

XF (REAL)
NONDIM FACTOR STAS
ENTER NX REAL VALUE(S)

(Enter the factor stations in ascending order; need not be coincident with component module blade stations or aerodynamic coefficient blade stations; must start with 0 and end with 1; 0 $\leq$ XF(I) $\leq$ 1)

FL (REAL)
FACTORS FOR CL
ENTER NX REAL VALUE(S)

(Enter the factor to be applied to CL at each factor station)

FD (REAL)
FACTORS FOR CD
ENTER NX REAL VALUE(S)

FM (REAL)
FACTORS FOR CM
ENTER NX REAL VALUE(S)

3.2.5 FFAØ - Fuselage Aerodynamics, Flat Plate Drag. FFAØ simulates a simple flat plate drag force parallel to the fuselage longitudinal axis.

3.2.5.1 Input, FFAØ - The system state vector is passed automatically at solution time. Air density ratio is input as a global variable during model formulation. The FFAØ input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS

response

VAIR (REAL)
WIND SPEED
(FT/SEC)
ENTER 1 REAL VALUES

(Enter the relative wind speed; positive in XCG direction)

CD (REAL)
TOTAL DRAG COEFF
ENTER 1 REAL VALUE

YANGL (REAL)
WND AXS-REF DOF ANGL
ENTER 1 REAL VALUE

(Enter the angle [deg] in the X-Z plane between the wind axis and the fuselage longitudinal axis ; $-90° \leq$ YANGL $\leq 90°$)

3.2.6 FFC2 - Fuselage Aerodynamics, Linear (2-d). FFC2 simulates the lift and drag forces and pitching moment on the wing, fuselage, horizontal tail, or vertical tail which may be associated with a fuselage component. Tail rotor and auxiliary propulsion forces may also be included. The resultant forces and moments at the fuselage CG are then calculated, but do not include the modal generalized forces (forces are applied to rigid body modes only).

3.2.6.1 Primary Features -

Quadratic Aerodynamic Coefficient Approximation - Wing, fuselage, horizontal tail, and vertical tail are considered to be 2-dimensional rectangular airfoils. C_L, C_D, and C_M are approximated by the partial series

$$C = C_0 + \left[\frac{\partial C}{\partial \alpha}\right]\alpha + \left[\frac{\partial^2 C}{\partial \alpha^2}\right]\alpha^2 = C_0 + C_1\alpha + C_2\alpha^2$$

where C may be C_L, C_D, or C_M. Note that the pitching moment is calculated using a reference length of 1 inch:

$$M = q\ Ac\ C_M$$

where q is the dynamic pressure, A is area, and c is the reference length.

3.2.6.2 Input, FFC2 - The system state vector is passed automatically at solution time. Air density ratio is input as a global variable during model formulation. The FFC2 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response

Only required input (based on previous input) is requested.

VAIR (REAL)
WIND VELOCITY
VX, VY, VZ (FT/SEC) W.R.T BODY SYSTEM
ENTER 3 REAL VALUES

(Enter the relative wind velocity vector with respect to the fuselage coordinate system)

IFUSE (Y OR N)
FUSELAGE AERODYNAMIC
FORCES
ENTER 1 Y OR N VALUE

(Enter Y if fuselage aerodynamic forces are to be included)

RFUSE (REAL)
FUSELAGE AERODYNAMIC
CNTR LOC
RX, RY, RZ IN TERMS OF X, Y, Z COORDINATE SYSTEM (IN)
ENTER 3 REAL VALUES

(Enter the fuselage aerodynamic center location with respect to the fuselage CG)

AFUSE (REAL)
FUSELAGE AREA
(SQR FT)
ENTER 1 REAL VALUE

(Enter the fuselage planform area; AFUS > Ø)

ALØFUS (REAL)
FUSELAGE LIFT COEFF
CØ
ENTER 1 REAL VALUE

(See paragraph 3.2.6.1, Quadratic Aerodynamic Coefficient Approximation)

AL1FUS (REAL)
FUSELAGE LIFT COEFF
C1 (1/DEG)
ENTER 1 REAL VALUE

AL2FUS (REAL)
FUSELAGE LIFT COEFF
C2 (1/DEG/DEG)
ENTER 1 REAL VALUE

CDØFUS (REAL)
FUSELAGE DRAG COEFF
CØ
ENTER 1 REAL VALUE

CD1FUS (REAL)
FUSELAGE DRAG COEFF
C1 (1/DEG)
ENTER 1 REAL VALUE

CD2FUS (REAL)
FUSELAGE DRAG COEFF
C2 (1/DEG/DEG)
ENTER 1 REAL VALUE

CMØFUS (REAL)
FUSELAGE MOM COEFF
CØ
ENTER 1 REAL VALUE

CM1FUS (REAL)
FUSELAGE MOM COEFF
C1 (1/DEG)
ENTER 1 REAL VALUE

CM2FUS (REAL)
FUSELAGE MOM COEFF
C2 (1/DEG/DEG)
ENTER 1 REAL VALUE

FUINCI (REAL)
FUSELAGE INCIDENCE
(DEG)
ENTER 1 REAL VALUE

(Enter the angle in the X-Z plane between the aerodynamic chord line and the fuselage longitudinal axis)

FUDO (REAL)
FUSELAGE DOWNWASH
ANGLE (DEG)
ENTER 1 REAL VALUE

(Enter the angle change in the relative wind velocity due to the fuselage wake)

FURATI (REAL)
FUSELAGE VELOCITY
RATIO
(LOCAL STREAM VEL/FREE STREAM VEL)**2
ENTER 1 REAL VALUE

(Enter $[V/V_\infty]^2$ of the fuselage)

IWING (Y OR N)
CONSIDER WING
ENTER 1 Y OR N VALUE

(Enter Y if wing aerodynamic forces are to be included)

RWING (REAL)
WING AERODYNAMIC
CNTR LOC
RX, RY, RZ IN TERMS OF X, Y, Z COORDINATE SYSTEM (IN)
ENTER 3 REAL VALUES

(Enter the wing aerodynamic center location with respect to the fuselage CG)

AWING (REAL)
WING AREA (SQR FT)
ENTER 1 REAL VALUE

(Enter the wing area [chord x span]; AWING > Ø)

ALØWIN (REAL)
WING LIFT COEFF CØ
ENTER 1 REAL VALUE

(See paragraph 3.2.6.1, Quadratic Aerodynamic Coefficient Approximation)

AL1WIN (REAL)
WING LIFT COEFF C1
(1/DEG)
ENTER 1 REAL VALUE

AL2WIN (REAL)
WING LIFT COEFF C2
(1/DEG/DEG)
ENTER 1 REAL VALUE

CDØWIN (REAL)
WING DRAG COEFF CØ
ENTER 1 REAL VALUE

CD1WIN (REAL)
WING DRAG COEFF C1
(1/DEG)
ENTER 1 REAL VALUE

CD2WIN (REAL)
WING DRAG COEFF C2
(1/DEG/DEG)
ENTER 1 REAL VALUE

CMØWIN (REAL)
WING MOM COEFF CØ
ENTER 1 REAL VALUE

CM1WIN (REAL)
WING MOM COEFF C1
(1/DEG)
ENTER 1 REAL VALUE

CM2WIN (REAL)
WING MOM COEFF C2
(1/DEG/DEG)
ENTER 1 REAL VALUE

WINCI (REAL)
WING INCIDENCE
(DEG)
ENTER 1 REAL VALUE

(Enter the angle in the X-Z plane between the airfoil chord line and the fuselage longitudinal axis)

WINGDO (REAL)
WING DOWNWASH ANGLE
(DEG)
ENTER 1 REAL VALUE

(Enter the angle change in the relative wind velocity due to the wing wake)

WRATI (REAL)
WING VELOCITY RATIO
(LOCAL STREAM VEL/FREE STREAM VEL)**2
ENTER 1 REAL VALUE

(Enter $[V/V_\infty]^2$ of the wing)

IHT (Y OR N)
CONSIDER HORIZONTAL
TAIL
ENTER 1 Y OR N VALUE

(Enter Y if horizontal tail aerodynamic forces are to be included)

RHT (REAL)
HTAIL AERODYNAMIC
CNTR LOC
RX, RY, RZ IN TERMS OF X, Y, Z COORDINATE SYSTEM (IN)
ENTER 3 REAL VALUES

(Enter the horizontal tail aerodynamic center location with respect to the fuselage CG)

AHT (REAL)
HORIZONTAL TAIL AREA
(SQR FT)
ENTER 1 REAL VALUE

(Enter the horizontal tail area [chord x span]; AHT > Ø)

ALØHT (REAL)
HTAIL LIFT COEFF CØ
ENTER 1 REAL VALUE

(See paragraph 3.2.6.1, Quadratic Aerodynamic Coefficient Approximation)

AL1HT (REAL)
HTAIL LIFT COEFF C1
(1/DEG)
ENTER 1 REAL VALUE

AL2HT (REAL)
HTAIL LIFT COEFF C2
(1/DEG/DEG)
ENTER 1 REAL VALUE

CDØHT (REAL)
HTAIL DRAG COEFF CØ
ENTER 1 REAL VALUE

CD1HT (REAL)
HTAIL DRAG COEFF C1
(1/DEG)
ENTER 1 REAL VALUE

CD2HT (REAL)
HTAIL DRAG COEFF C2
(1/DEG/DEG)
ENTER 1 REAL VALUE

CMØHT (REAL)
HTAIL MOM COEFF CØ
ENTER 1 REAL VALUE

CM1HT (REAL)
HTAIL MOM COEFF C1
(1/DEG)
ENTER 1 REAL VALUE

CM2HT (REAL)
HTAIL MOM COEFF C2
(1/DEG/DEG)
ENTER 1 REAL VALUE

HTINCI (REAL)
HTAIL INCIDENCE
(DEG)
ENTER 1 REAL VALUE

(Enter the angle in the X-Z plane between the airfoil chord line and the fuselage longitudinal axis)

HTDO (REAL)
HTAIL DOWNWASH ANGLE
(DEG)
ENTER 1 REAL VALUE

(Enter the angle change in the relative wind velocity due to the horizontal tail wake)

HTRATI (REAL)
HTAIL VELOCITY RATIO
(LOCAL STREAM VEL/FREE STREAM VEL)**2
ENTER 1 REAL VALUE

(Enter $[V/V_\infty]^2$ of the horizontal tail)

IVT (Y OR N)
CONSIDER VERTICAL
TAIL
ENTER 1 Y OR N VALUE

(Enter Y if vertical tail aerodynamic forces are to be included)

RVT (REAL)
VTAIL AERODYNAMIC
CNTR LOC
RX, RY, RZ IN TERMS OF X, Y, Z COORDINATE SYSTEM (IN)
ENTER 3 REAL VALUES

(Enter the vertical tail aerodynamic center location with respect to the fuselage CG)

AVT (REAL)
VERTICAL TAIL AREA
(SQR FT)
ENTER 1 REAL VALUE

(Enter the vertical tail area [chord x span]; AVT > Ø)

ALØVT (REAL)
VTAIL LIFT COEFF CØ
ENTER 1 REAL VALUE

(See paragraph 3.2.6.1, Quadratic Aerodynamic Coefficient Approximation)

AL1VT (REAL)
VTAIL LIFT COEFF C1
(1/DEG)
ENTER 1 REAL VALUE

AL2VT (REAL)
VTAIL LIFT COEFF C2
(1/DEG/DEG)
ENTER 1 REAL VALUE

CDØVT (REAL)
VTAIL DRAG COEFF CØ
ENTER 1 REAL VALUE

CD1VT (REAL)
VTAIL DRAG COEFF C1
(1/DEG)
ENTER 1 REAL VALUE

CD2VT (REAL)
VTAIL DRAG COEFF C2
(1/DEG/DEG)
ENTER 1 REAL VALUE

CMØVT (REAL)
VTAIL MOM COEFF CØ
ENTER 1 REAL VALUE

CM1VT (REAL)
VTAIL MOM COEFF C1
(1/DEG)
ENTER 1 REAL VALUE

CM2VT (REAL)
VTAIL MOM COEFF C2
(1/DEG/DEG)
ENTER 1 REAL VALUE

VTINCI (REAL)
VTAIL INCIDENCE
(DEG)
ENTER 1 REAL VALUE

(Enter the angle in the X-Y plane between the airfoil chord line and the fuselage longitudinal axis)

VTDO (REAL)
VTAIL DOWNWASH ANGLE
(DEG)
ENTER 1 REAL VALUE

(Enter the angle change in the relative wind velocity due to the vertical tail wake)

VTRATI (REAL)
VTAIL VELOCITY RATIO
(LOCAL STREAM VEL/FREE STREAM VEL)**2
ENTER 1 REAL VALUE

(Enter $[V/V_\infty]^2$ of the vertical tail)

ITAIL (Y OR N)
CONSIDER TAIL ROTOR
ENTER 1 Y OR N VALUE

(Enter Y if tail rotor thrust is to be included)

RTAIL (REAL)
TAIL ROTOR LOCATION
RX, RY, RZ IN TERMS OF X, Y, Z COORDINATE SYSTEM (IN)
ENTER 3 REAL VALUES

(Enter the tail rotor location with respect to the fuselage CG)

TTAIL (REAL)
TAIL ROTOR THRUST
ENTER 1 REAL VALUE

(Enter thrust [lb]; positive TTAIL produces a positive moment about the fuselage Z-axis)

IPROP (Y OR N)
CONSIDER PROPELLER
ENTER 1 Y OR N VALUE

(Enter Y if propeller thrust is to be included)

RPROP (REAL)
PROPELLER LOCATION
RX, RY, RZ IN TERMS OF X, Y, Z COORDINATE SYSTEM (IN)
ENTER 3 REAL VALUES

(Enter the propeller location with respect to the fuselage CG)

TPROP (REAL)
PROPELLER THRUST
ENTER 1 REAL VALUE

(Enter thrust [lb]; positive TPROP produces a force in the positive X direction [see Figure 13])

3.2.7 FLA2 - Lifting Surface Aerodynamics. FLA2 computes the lift, pitching moment, rolling moment, and induced drag on the modal lifting surface (paragraph 3.1.15) using subsonic lifting surface theory for finite wings.

3.2.7.1 Primary Features -

Steady Aerodynamics - The mean cambered surface is represented with a doublet sheet. The doublet strength is computed with the influence of all other sections taken into account. Euler's equation is

then used to solve for the downwash (induced angle of attack) at each section resulting from the loading distribution.

Semispan Representation - Lifting surfaces may be composed of one or two (full wing) semi-spans and may include sweep and taper.

Control Surface - A control surface option is included (limited to symmetric or antisymmetric deflections when used with two semi-spans).

Section Properties - Section and control surface lift and moment curves (constant slope) are required.

3.2.7.2 Input, FLA2 - The FLA2 input prompts are shown below in capital letters. Input prompt format is:

VARIABLE NAME (VARIABLE TYPE)
VARIABLE DESCRIPTION
REQUIRED NUMBER AND TYPE OF INPUTS
response

Only required input (based on previous input) is requested.

NWNG (INTEGER)
NO. OF SEMI-SPANS
ENTER 1 INTEGER VALUE

(A full wing is defined as two equal-length semi-spans; $1 \leq$ NWNG ≤ 2)

B2 (REAL)
LENGTH OF SEMI-SPAN
(IN)
ENTER 1 REAL VALUE

S (REAL)
PLANFORM AREA
(IN**2)
ENTER 1 REAL VALUE

(Enter total planform area)

TR (REAL)
TAPER RATIO
(TIP CHORD/ROOT CHORD)
ENTER 1 REAL VALUE

(TR > Ø)

QMACH (REAL)
FREE STREAM MACH NO.
ENTER 1 REAL VALUE

(Ø < QMACH < 1.0)

QLAM (REAL)
LEADING EDGE SWEEP
(DEGREES)
ENTER 1 REAL VALUE

(Ø ≤ QLAM < 90)

ICS (Y OR N)
CONTROL SURFACE
ENTER 1 Y OR N VALUE

YCSO (REAL)
SPANWISE LOCATION
CONTROL SURFACE OUTBD EDGE (PERCENT SEMI-SPAN)
ENTER 1 REAL VALUE

(Enter the spanwise location of the outboard edge of the control surface as a percentage of the semi-span length; YCSO > Ø)

YCSI (REAL)
SPANWISE LOCATION
CONTROL SURFACE INBD EDGE (PERCENT SEMI-SPAN)
ENTER 1 REAL VALUE

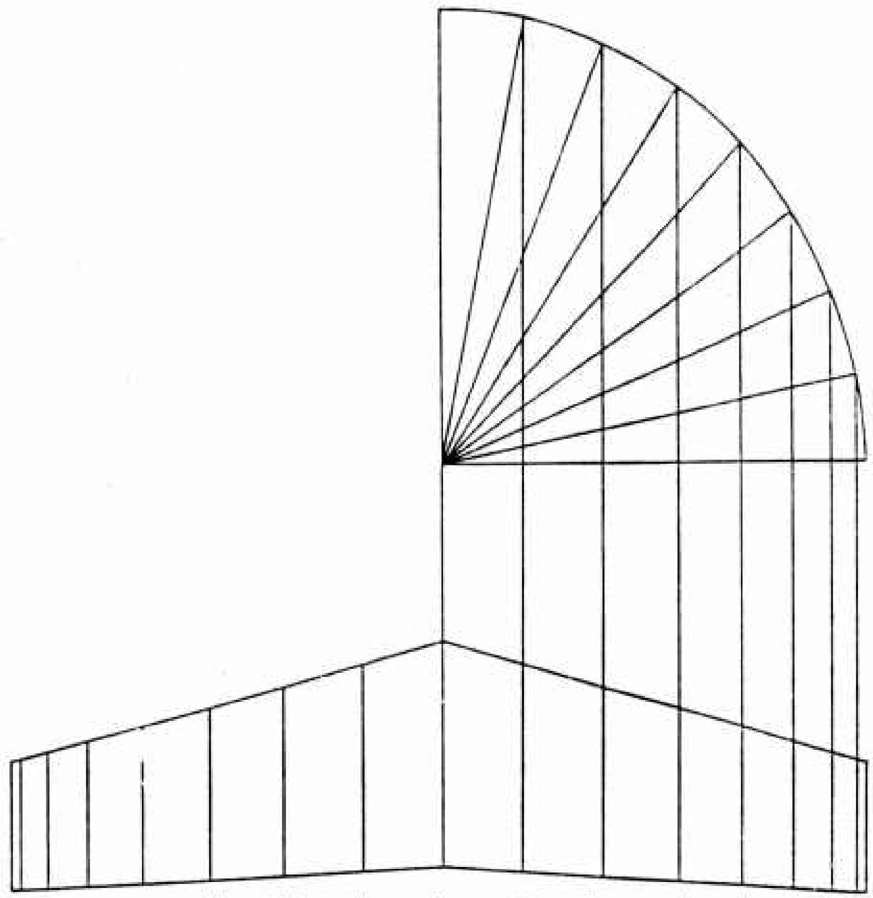
(The inboard edge of the control surface is constrained to be at the lifting surface root unless 2 semi-spans (full wing) are specified; YCSI < YCSO)

ISYM (INTEGER)
CONTROL SYMMETRY
1 = SYMMETRICAL CONTROL DEFLECTION
-1 = ANTISYMMETRICAL CONTROL DEFLECTION
ENTER 1 INTEGER VALUE

(Control deflection is limited to symmetric or antisymmetric displacements when 2 semi-spans have been specified; CLC- must be used to impose this constraint on the control surface degrees of freedom)

CLA8 (REAL)
LIFT CURVE SLOPE
(PER RAD)
ENTER 8 REAL VALUES

(Enter the section lift curve slopes - single semi-span; 8 spanwise stations (15 if 2 semi-spans) are defined as shown in Figure 22)



STATION NO.	NONDIMENSIONAL STATION
1	0.
2	0.1951
3	0.3827
4	0.5556
5	0.7071
6	0.8315
7	0.9239
8	0.9808

Figure 22. Definition of Spanwise Stations.

AØ8 (REAL)
Ø LIFT ANGLE
(DEG)
ENTER 8 REAL VALUES

CMA8 (REAL)
MOMENT CURVE SLOPE
(PER RAD)
ENTER 8 REAL VALUES

CMØ8 (REAL)
Ø LIFT MOMENT
COEFFICIENT
ENTER 8 REAL VALUES

CLA15 (REAL)
LIFT CURVE SLOPE
(PER RAD)
ENTER 15 REAL VALUES

(Enter the section lift curve slopes - two semi-spans; 15 spanwise stations (8 if single semi-span) are defined as shown in Figure 22.

AØ15 (REAL)
Ø LIFT ANGLE
(DEG)
ENTER 15 REAL VALUES

CMA15 (REAL)
MOMENT CURVE SLOPE
(PER RAD)
ENTER 15 REAL VALUES

CMØ15 (REAL)
Ø LIFT MOMENT
COEFFICIENT
ENTER 15 REAL VALUES

CLD (REAL)
C S LIFT CURVE
SLOPE (PER RAD)
ENTER 1 REAL VALUE

(Enter the control surface lift curve slope)

CMD (REAL)
C S MOMENT CURVE
SLOPE (PER RAD)
ENTER 1 REAL VALUE

PA (REAL)
AMBIENT PRESSURE
(PSI)
ENTER 1 REAL VALUE

3.3 SOLUTION TECHNOLOGY MODULES

A solution module represents an algorithm for computing eigenvalue, time domain, or frequency domain solutions for a given formulation of a set of second order differential equations.

During interactive input, the FORTRAN name of the parameter and a descriptive message are displayed. Only required data, based on previous input, is requested. Certain parameters have range constraints. If these are not satisfied, a message will be displayed requesting a reentry. In certain cases, the user will have an option to override the formal constraint.

3.3.1 SEA3 - Eigenanalysis. SEA3 computes the eigenvalues and eigenvectors of the constant M and K matrices of a system using the power method. A solution will not be computed if rigid body modes are present (singular K matrix).

3.3.1.1 Input, SEA3 - The SEA3 input prompts are shown below in capital letters. Only required input (based on previous input) is requested.

NMODES (INTEGER)
NO. OF MODES
ENTER 1 INTEGER VALUE

(Enter the number of modes, n, for which the eigenvalues and eigenvectors will be output; n must be greater than zero and less than or equal to the number of system degrees of freedom; the first n modes are output)

ITMAX (INTEGER)
MAX NO. ITERATIONS
ENTER 1 INTEGER VALUE

(Enter the maximum number of iterations to be allowed for each mode; if the ratio tolerances have not been satisfied within the given number of iterations, the solution is terminated; the maximum number of iterations must be greater than or equal to 20)

TOL (REAL)
RATIO TOLERANCE
EIGENVALUE

(The ratio tolerance defines the numeric bounds for the computed eigenvalues; the ratio tolerance must be less than or equal to .0001)

TOLM (REAL)
RATIO TOLERANCE
EIGENVECTOR

(See TOL)

3.3.1.2 Output, SEA3 - For each mode:

Frequency (Hz)

Frequency (rad/sec)

Generalized mass ($lb\text{-}sec^2/in.$) - mass

($lb\text{-}sec^2\text{-}in.^2/in.$) - mass moment of inertia

System degrees of freedom

Normalized eigenvector

If a negative eigenvalue is calculated (M or K not positive definite), the solution is terminated.

3.3.2 SEA4 - Eigenanalysis. SEA4 computes the eigenvalues and eigenvectors, including rigid body modes, of the constant M and K matrices of a system using the Householder method.

3.3.2.1 Input, SEA4 - The SEA4 input prompts are shown below in capital letters. Only required input (based on previous input) is requested.

NMODES (INTEGER)
NUMBER OF MODES
ENTER 1 INTEGER VALUE

(Enter the number of modes, n, for which the eigenvalues and eigenvectors will be output; n must be greater than zero and less than or equal to the number of system degrees of freedom; the first n modes are output)

3.3.2.2 Output, SEA4 - For each mode:

Frequency (Hz)
Frequency (rad/sec)
Generalized mass ($\text{lb-sec}^2/\text{in.}$) - mass
($\text{lb-sec}^2\text{-in.}^2/\text{in.}$) - mass moment of inertia
System degrees of freedom
Normalized eigenvector

Small negative values may be computed for rigid body eigenvalues due to numerical inaccuracy; these are automatically set to zero and a warning issued. If M is not positive definite or if K is not positive semi-definite, the solution is terminated.

3.3.3 SEA5 - Eigenanalysis. SEA5 computes the complex eigenvalues and eigenvectors, including rigid body modes, of the constant M, C, and K matrices of a system using a generalized Householder algorithm. The user can select the component and system degrees of freedom for which eigenvectors are to be output.

3.3.3.1 Input, SEA5 - The SEA5 input prompts are shown below in capital letters. Only required input (based on previous input) is requested.

NMODES (INTEGER)
NUMBER OF MODES
ENTER 1 INTEGER VALUE

(Enter the number of modes for which the eigenvalues and eigenvectors will be output; $1 \leq \text{NMODES} \leq 2 \cdot \text{NSDF}$)

IDAMP (Y OR N)
CONSIDER DAMPING
MATRIX
ENTER 1 Y OR N VALUE

DOFPRINT (MODEL DOFS CHOSEN)
DOFS TO BE PRINTED

(The indexed system degree of freedom names are listed)

ALL SYSTEM DOFS (Y OR N)

(Enter Y if the eigenvectors for all system degrees of freedom are to be output, else N)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the indices of the system degrees of freedom for which the eigenvectors will be output, end with Ø)

ANY COMPONENT DOFS (Y OR N)

(Enter Y if the eigenvectors of any model components are to be output, else N; if Y the indexed component data set/data member names are listed)

SELECT COMPONENTS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the component indices, end with Ø)

For each component:

DOFS FOR COMPONENT ds/dm

(The indexed component degree of freedom names are listed)

SELECT DOF INDICES (Y OR N) OR QUIT COMPONENTS (Q)

(Enter Y if any degrees of freedom will be selected from this component, else N, continue with next component, else Q, terminate all component degree of freedom selection)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES-TO TERMINATE ENTER Ø

(Enter the indices of the component degrees of freedom for which the eigenvectors will be output, end with Ø)

3.3.3.2 Output, SEA5 - Output is the system eigenvalues and the system/component normalized displacement and velocity eigenvectors. The system eigenvalues occur as complex conjugate pairs and are output as frequency (Hz) and damping (lb-sec/in., lb-sec-in./rad) pairs. A positive value for damping indicates an unstable system.

3.3.4 STH3 - Time History. STH3 performs a Runge-Kutta integration of the system equations with generalized initial conditions and an optional error

check. If the error check is elected, the integration increment is automatically halved until the specified accuracy is achieved. The user also has the option to specify separate integration increments for time-dependent system coefficients and forces.

Helicopter rotor control parameters can be input and the computed aerodynamic forces and torque on the rotor hub can optionally be output. A reference global coordinate system in which gravitational and centrifugal acceleration vectors can be specified may be defined (see paragraph 3.4). Time history state vectors can optionally be saved for interface and internal loads calculations (see paragraphs 2.1 and 3.3.9). In addition, the final conditions of a system from a previous run can be used as initial conditions in order to continue a time history solution. The model can be different for the continued run if the system degrees of freedom are the same.

3.3.4.1 Input, STH3 - The STH3 input prompts are shown below in capital letters. Only required input (based on previous input) is requested.

TSTA (REAL)
START TIME (SEC)
ENTER 1 REAL VALUE

(Enter Ø unless continuing a previous run; TSTA ≥ Ø)

H (REAL)
INITIAL INCREMENT
(SEC)
ENTER 1 REAL VALUE

(Enter the initial integration increment; H ≥ Ø)

HTD (REAL)
SEPARATE INCREMENT
TIME DEPT COEFFS
ENTER 1 REAL VALUE

(Separate integration increments may be entered, one each for time-dependent system coefficients and time-dependent system forces; if either input is nonzero, no error check will be performed; if an input is negative, the associated time-dependent parameters will not be computed; if zero is input, a separate increment will not be used)

HF (REAL)
SEPARATE INCREMENT
FORCE COMPUTATION
ENTER 1 REAL VALUE

TEND (REAL)
END TIME (SEC)
ENTER 1 REAL VALUE

(TEND > TSTA)

E (REAL)
ERROR CHECK VALUE
IF Ø, THEN CONSTANT INCREMENT USED
ENTER 1 REAL VALUE

(The error check is applied to a system degree of freedom chosen by the user; the value entered for the error check defines the bounds for the computed displacement of the degree of freedom after each integration; the integration increment is automatically halved until the specified accuracy is achieved; if zero is input, the error check is omitted, which is more efficient computationally, but may cause inaccuracies or numerical instability; displacement [in.], angular displacement [rad]; Ø < E ≤ .0001)

IDFLI (SYSTEM DOFS CHOSEN)
TEST DOF

(The indexed system degree of freedom names are listed)

SELECT ONE SYSTEM DOF BY INDEX

(Enter the index of the degree of freedom to be used for the error check)

ICOPT (INTEGER)
INITIAL CONDITION
INPUT TYPE
Ø = NONE, 1 = SINGLE DISPLACEMENT, 2 = GENERAL, 3 = CONTINUE
ENTER 1 INTEGER VALUE

if Ø:

no initial condition input, all initial conditions set to zero

if 1:

enter initial condition; displacement (in.), angular displacement (rad)
enter name of degree of freedom for initial condition

if 2:

enter velocity and displacement for each system degree of freedom

if 3:
no input, initial conditions will be final values from previous time history

VI (REAL)
INITIAL DISPLACEMENT
SINGLE SYSTEM DOF (IN OR RAD)
ENTER 1 REAL VALUE

IIDFLI (SYSTEM DOFS CHOSEN)
DOF NAME
INITIAL CONDITION

(The indexed system degree of freedom names are listed)

SELECT ONE SYSTEM DOF BY INDEX

(Enter the index of the degree of freedom at which the initial condition will be applied)

YV (REAL)
INITIAL VELOCITY
EACH SYSTEM DOF
ENTER NSDF REAL VALUES

YD (REAL)
INITIAL DISPLACEMENT
EACH SYSTEM DOF
ENTER NSDF REAL VALUES

ICON (INTEGER)
ROTOR CONTROLS
INPUT TYPE Ø = NULL, 1 = GENERAL, 2 = CONTINUE
ENTER 1 INTEGER VALUE

if Ø :

no rotor controls input

if 2:

no input, control inputs will be values from previous time history

if 1:

MROT (INTEGER)
NUMBERS OF ROTORS
ENTER 1 INTEGER VALUE

(Enter the number of rotors at which controls will be applied: $1 \leq \text{MROT} \leq 4$)

for each rotor:

IRi (INTEGER)
ROTOR NUMBER
ENTER 1 INTEGER VALUE

(Enter the rotor number of the ith rotor for which there will be control inputs)

AØi (REAL)
COLLECTIVE ANGLE (RAD)
ENTER 1 REAL VALUE

(Rotor fundamental control inputs are of the form

$$\theta_\emptyset + \theta_1 \cos \Omega t + \theta_2 \sin \Omega t$$

where $\theta_\emptyset$ is the collective input, θ_1 and θ_2 are the cyclic input components, and Ωt is the azimuth of blade 1; enter $\theta_\emptyset$)

A1Ci (REAL)
COSINE ANGLE (RAD)
ENTER 1 REAL VALUE

(Enter θ_1)

A1Si (REAL)
SINE ANGLE (RAD)
ENTER 1 REAL VALUE

(Enter θ_2)

NHi (INTEGER)
HIGHER HARMONIC
COLLECTIVE HARMONIC, Ø = NONE
ENTER 1 INTEGER VALUE

(Rotor higher harmonic control inputs are of the form

$$\theta_3 \cos n\Omega t + \theta_4 \sin n\Omega t$$

where θ_3 and θ_4 are higher harmonic collective input components, Ωt is the azimuth of blade 1, and n is the harmonic; enter n)

ANCi (REAL)
COS ANGLE (RAD)
ENTER 1 REAL VALUE

(Enter θ_3)

ANSi (REAL)
SIN ANGLE (RAD)
ENTER 1 REAL VALUE

(Enter θ_4)

ITOUT (INTEGER)
ROTOR FORCE OUTPUT
NO. OF ROTORS
ENTER 1 INTEGER VALUE

(Enter the number of rotors for which the computed aerodynamic forces and torque on the rotor hub are to be output; hub forces are along the principal axes of the hub fixed coordinate system)

ITROT (INTEGER)
ROTOR NUMBERS
ENTER ITOUT INTEGER VALUES

CRT (Y OR N)
OUTPUT (at) THIS TERMINAL
ENTER 1 Y OR N VALUE

OUTTH (INTEGER)
OUTPUT DEVICE
NO. FOR TIME HISTORY SOLUTION
ENTER 1 INTEGER VALUE

ITEET (Y OR N)
TEETERING ROTOR
ENTER 1 Y OR N VALUE

LINK (REAL)
HORIZONTAL TAIL LINK
GIVE A, B, C VALUES SO THAT THE CYCLIC SINE CONTROL RELATES TO THE HORIZONTAL TAIL INCIDENCE IN THE FOLLOWING WAY:
INCIDENCE ANGLE = A + B*A1S + (C*A15)**2
ENTER 3 REAL VALUES

(This relationship is imposed in order to provide effective fuselage moments for improved control performance with teetering rotors)

ILOP (Y OR N)
SAVE STATE VECTORS
FOR INTERFACE, INTERNAL LOADS CALCULATIONS
ENTER 1 Y OR N VALUE

(Enter Y if time history state vectors are to be saved for loads calculations; see paragraphs 2.1 and 3.3.9)

JIIL (INTEGER)
INPUT I, EVERY ITH
STATE VECTOR TO BE WRITTEN TO LOADS FILE

$(1 \leq \text{JIIL} \leq 10)$

GS (REAL)
VERT ACCELERATION
(G/GØ)
ENTER 1 REAL VALUE

(Enter the vertical acceleration in the inertial system in g's; else zero if global reference system not used; see paragraph 3.4)

ICF (Y OR N)
CONSIDER CENTRIFUGAL
ACCELERATION
ENTER 1 Y OR N VALUE

(Enter Y if centrifual acceleration due to steady turn is to be considered, else N; see paragraph 3.4)

OMG (REAL)
TURN RATE
(DEG/SEC)
ENTER 1 REAL VALUE

(OMG ≥ Ø)

RT (REAL)
RADIUS OF TURN
(FT)
ENTER 1 REAL VALUE

(RT ≥ Ø)

XYDI (REAL)
GLOBAL X,Y VECTORS
IN TERMS OF INERTIAL SYSTEM
ENTER 6 REAL VALUES

(Enter the X and Y vectors of the global coordinate system in terms of the inertial coordinate system [X cross Y yields the Z vector]; null vector if global reference system not used; see Figure 19 and paragraph 3.4)

3.3.4.2 Output, STH3 - The system degree of freedom displacements (and rotor hub aerodynamic forces and torque) are printed at each time step, and the state vector is printed at the end of the run.

3.3.5 STH4 - Time History. STH4 performs a Runge-Kutta integration of the system equations with generalized initial conditions and an optional error check. If the error check is elected, the integration increment is automatically halved until the specified accuracy is achieved. The user also has the option to specify separate integration increments for the time-dependent system coefficients and forces.

Rotor control input and rotor force output options are not available. A reference global coordinate system, in which gravitational and centrifugal acceleration vectors can be specified, may be defined (see paragraph 3.4). The user can select the component and system degrees of freedom for which the state vector or, separately, displacement or velocity will be output at each time increment. Time history output can optionally be written to an attached plot file which can be post processed for plotting or other purposes. State vectors can optionally be saved for interface and internal loads calculations (see paragraphs 2.1 and 3.3.9).

Provision has been made for coded flags to be issued automatically with the interactive output when specific dynamic conditions are encountered by or in effect for a given component. Augments can be added to a component Active Module (C---A) as shown below. The user can add, delete, or change conditions as desired without affecting output format.

```
SUBROUTINE FCT
      .
      .
      .
CALL C---A (..., ANFLAG)
SUBROUTINE C---A (..., ANFLAG)
      .
      .
      .
INTEGER ANFLAG, CODE1, CODE2, ..., CODEn [n≤40]
INCLUDE (FLAG) [syntax is machine dependent]
DATA CODE1/'AAAA'/,CODE2/'BBBB'/,..., CODEn/'xxxx'/
      .
      .
      .
ANFLAG = Ø
```

```
IF (CONDITION):
ANFLAG = ANFLAG + 1
ACFLAG (ANFLAG) = CODEn
```

The final conditions of a system from a previous run can be used as initial conditions in order to continue a time history solution. The model can be different for the continued run if the system degrees of freedom are the same.

3.3.5.1 Input, STH4 - The STH4 input prompts are shown below in capital letters. Only required input (based on previous input) is requested.

TSTA (REAL)
START TIME (SEC)
ENTER 1 REAL VALUE

(Enter Ø unless continuing a previous run; TSTA ≥ Ø)

H (REAL)
INITIAL INCREMENT
(SEC)
ENTER 1 REAL VALUE

(Enter the initial integration increment; H ≥ Ø)

HTD (REAL)
SEPARATE INCREMENT
TIME DEP COEFFS
ENTER 1 REAL VALUE

(Separate integration increments may be entered, one each for time-dependent system coefficients and time-dependent system forces; if either input is nonzero, no error check will be performed; if an input is negative, the associated time-dependent parameters will not be computed; if zero is input, a separate increment will not be used)

HF (REAL)
SEPARATE INCREMENT
FORCE COMPUTATION
ENTER 1 REAL VALUE

TEND (REAL)
END TIME (SEC)
ENTER 1 REAL VALUE

(TEND > TSTA)

E (REAL)
ERROR CHECK VALUE
IF Ø, THEN CONSTANT INCREMENT USED
ENTER 1 REAL VALUE

(The error check is applied to a system degree of freedom chosen by the user; the value entered for the error check defines the bounds for the computed displacement of the degree of freedom after each integration; the integration increment is automatically halved until the specified accuracy is achieved; if zero is input, the error check is omitted, which is more efficient computationally, but may cause inaccuracies or numerical instability; [in], angular displacement [rad])

IDFLI (SYSTEM DOFS CHOSEN)
TEST DOF

(The indexed system degree of freedom names are listed)

SELECT ONE SYSTEM DOF BY INDEX

(Enter the index of the degree of freedom to be used for the error check)

ICOPT (INTEGER)
INITIAL CONDITION
INPUT TYPE
Ø = NONE, 1 = SINGLE DISPLACEMENT, 2 = GENERAL, 3 = CONTINUE
ENTER 1 INTEGER VALUE

if Ø:

no initial condition input, all initial conditions set to zero

if 1:

enter initial condition; displacement (in.), angular displacement (rad)
enter name of degree of freedom for initial condition

if 2:

enter velocity and displacement for each system degree of freedom

if 3:

no input, initial conditions will be final values from previous time history

VI (REAL)
INITIAL DISPLACEMENT
SINGLE SYSTEM DOF (IN OR RAD)
ENTER 1 REAL VALUE

IIDFLI (SYSTEM DOFS CHOSEN)
DOF NAME
INITIAL CONDITION

(The indexed system degree of freedom names are listed)

SELECT ONE SYSTEM DOF BY INDEX

(Enter the index of the degree of freedom at which the initial condition will be applied)

YV (REAL)
INITIAL VELOCITY
EACH SYSTEM DOF
ENTER NSDF REAL VALUES

YD (REAL)
INITIAL DISPLACEMENT
EACH SYSTEM DOF
ENTER NSDF REAL VALUES

CRT (Y OR N)
OUTPUT (at) THIS TERMINAL
ENTER 1 Y OR N VALUE

OUTTH (INTEGER)
OUTPUT DEVICE
NO. FOR TIME HISTORY SOLUTION
ENTER 1 INTEGER VALUE

PROP (INTEGER)
PRINT
1 = DISPLACEMENT 2 = VELOCITY 3 = BOTH
ENTER 1 INTEGER VALUE

DOFPRINT (MODEL DOFS CHOSEN)
DOFS TO BE PRINTED

(The indexed system degree of freedom names are listed)

ALL SYSTEM DOFS (Y OR N)

(Enter Y if there is to be output for all system degrees of freedom, else N)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the indices of the system degrees of freedom for which there will be output, end with Ø)

ANY COMPONENT DOFS (Y OR N)

(Enter Y if there is to be output for any model components, else N; if Y the indexed component data set/data member names are listed)

SELECT COMPONENTS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the component indices, end with Ø)

For each component:

DOFS FOR COMPONENT ds/dm

(The indexed component degree of freedom names are listed)

SELECT DOF INDICES (Y OR N) OR QUIT COMPONENTS (Q)

(Enter Y if any degrees of freedom will be selected from this component, else N, continue with next component, else Q, terminate all component degree of freedom selection)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the indices of the component degrees of freedom for which there will be output, end with Ø)

PLOP (INTEGER)
PLOT
Ø = NONE 1 = DISPLACEMENT 2 = VELOCITY 3 = BOTH
ENTER 1 INTEGER VALUE

(See paragraphs 2.1 and 3.3.5)

JPLT (INTEGER)
INPUT N, EVERY NTH
SOLUTION TO BE WRITTEN TO PLOT FILE
ENTER 1 INTEGER VALUE

($1 \leq$ JPLT ≤ 10)

DOFPLOT (MODEL DOFS CHOSEN)
DOFS TO BE PLOTTED

(The indexed system degree of freedom names are listed)

ALL SYSTEM DOFS (Y OR N)

(Enter Y if there is to be plot output for all system degrees of freedom, else N)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the indices of the system degrees of freedom for which there will be plot output, end with Ø)

ANY COMPONENT DOFS (Y OR N)

(Enter Y if there is to be plot output for any model components, else N; if Y, the indexed component data set/data member names are listed)

SELECT COMPONENTS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the component indices, end with Ø)

For each component:

DOFS FOR COMPONENT ds/dm

(The indexed component degree of freedom names are listed)

SELECT DOF INDICES (Y OR N) OR QUIT COMPONENTS (Q)

(Enter Y if any degrees of freedom will be selected from this component, else N, continue with next component, else Q, terminate all component degree of freedom selection)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the indices of the component degrees of freedom for which there will be plot output, end with Ø)

FLOP (Y OR N)
CONDITION CODES
TO BE OUTPUT
ENTER 1 Y OR N VALUE

ILOP (Y OR N)
SAVE STATE VECTORS
FOR INTERFACE, INTERNAL LOADS CALCULATIONS
ENTER 1 Y OR N VALUE

(Enter Y if time history state vectors are to be saved for loads calculations; see paragraphs 2.1 and 3.3.9)

JIIL (INTEGER)
INPUT I, EVERY ITH
STATE VECTOR TO BE WRITTEN TO LOADS FILE

($1 \leq$ JIIL ≤ 10)

GS (REAL)
VERT ACCELERATION
(G/GØ)
ENTER 1 REAL VALUE

(Enter the vertical acceleration in the inertial system in g's; else zero if global reference system not used; see paragraph 3.4)

ICF (Y OR N)
CONSIDER CENTRIFUGAL
ACCELERATION
ENTER 1 Y OR N VALUE

(Enter Y if centrifual acceleration due to steady turn is to be considered, else N; see paragraph 3.4)

OMG (REAL)
TURN RATE
(DEG/SEC)
ENTER 1 REAL VALUE

(OMG ≥ Ø)

RT (REAL)
RADIUS OF TURN
(FT)
ENTER 1 REAL VALUE

(RT ≥ Ø)

XYDI (REAL)
GLOBAL X,Y VECTORS
IN TERMS OF INERTIAL SYSTEM
ENTER 6 REAL VALUES

(Enter the X and Y vectors of the global coordinate system in terms of the inertial coordinate system [X cross Y yields the Z vector]; null vector if global reference system not used; see Figure 19 and paragraph 3.4)

3.3.5.2 <u>Output, STH4</u> - The component and system displacements and/or velocities are printed at each time step, and the system state vector is printed at the end of the run.

3.3.6 <u>SSF3 - Stability Floquet</u>. SSF3 uses periodic shooting to determine initial conditions for which a Runge-Kutta integration of general linear or nonlinear system equations is performed. An optional error check may be implemented in order to automatically adjust the integration increment until the specified accuracy is achieved. The user also has the option to specify separate integration increments for time-dependent system coefficients and forces. Helicopter rotor control parameters may also be input.

New initial conditions are determined and the integration repeated until the periodic equilibrium condition has been achieved or a specified number of

iterations has been exceeded. Periodic equilibrium is determined through error bounds on the system state vector. Perturbation of the periodic equilibrium state is then performed by the n-pass method in order to form the Floquet transition matrix. Finally, an eigenanalysis of the Floquet transition matrix is performed, from which the stability of the system may be evaluated.

3.3.6.1 Input, SSF3 - The SSF3 input prompts are shown below in capital letters. Only required input (based on previous input) is requested.

H (REAL)
INITIAL INCREMENT
(SEC)
ENTER 1 REAL VALUE

(Enter the initial integration increment; H ≥ Ø)

TPER (REAL)
INTEGRATION PERIOD
ENTER 1 REAL VALUE

(Time period [sec] for integration; enter the rotation period or forcing function period where applicable or an integer multiple of the integration increment [linear nonperiodic systems])

HTD (REAL)
SEPARATE INCREMENT
TIME DEP COEFFS
ENTER 1 REAL VALUE

(Separate integration increments may be entered, one each for time-dependent system coefficients and time-dependent system forces; if either input is nonzero, no error check will be performed; if an input is negative, the associated time-dependent parameters will not be computed; if zero is input, a separate increment will not be used)

HF (REAL)
SEPARATE INCREMENT
FORCE COMPUTATION
ENTER 1 REAL VALUE

E (REAL)
ERROR CHECK VALUE
IF Ø, THEN CONSTANT INCREMENT USED
ENTER 1 REAL VALUE

(The value entered for the error check defines the bounds for the computed state vector after each integration; the integration increment is automatically halved or doubled until the specified accuracy is achieved; if zero is input, the error check is omitted, which is more efficient computationally but may cause inaccuracies or numerical instability)

NALLOW (INTEGER)
NO. OF ITERATIONS
ALLOWED
ENTER 1 INTEGER VALUE

(If the periodic equilibrium state is not reached within the specified number of iterations, the solution is terminated; $1 \leq$ NALLOW $\leq$ 1000)

CEA (Y OR N)
CONSTANT ERROR
ALLOWED
ENTER 1 Y OR N VALUE

if YES:

Enter the value for the error allowed in the elements of the state vector; the same value is applied to the velocity and displacement of all system degrees of freedom; the allowed error establishes bounds for the determination of the periodic equilibrium state

if NO:

Enter the allowed error for the velocity and displacement of each system degree of freedom

CEALLO (REAL)
CONSTANT ERROR
ALLOWED
ENTER 1 REAL VALUE

(Ø < CEALLO $\leq$ .0001)

EALLOWV (REAL)
VELOCITY ERRORS
INPUT ERROR FOR EACH DOF
ENTER NSDF REAL VALUES

(Ø < EALLOWV(I) $\leq$ .0001)

EALLOWD (REAL)
DISPLACEMENT ERRORS
INPUT ERROR FOR EACH DOF
ENTER NSDF REAL VALUES

(Ø < EALLOWD(I) ≤ .0001)

CI (Y OR N)
CONSTANT INCREMENT
FOR COMPUTING TRANSITION MATRIX
ENTER 1 Y OR N VALUE

if YES:

Enter the increment to be applied to the state vector for the perturbation of initial conditions (the periodic equilibrium state); the same increment is used for the velocity and displacement of all system degrees of freedom

if NO:

Enter the increment for the velocity and displacement of each system degree of freedom

CICRE (REAL)
CONSTANT INCREMENT
TO BE USED
ENTER 1 REAL VALUE

(CICRE > Ø)

DELTAV (REAL)
VELOCITY INCREMENTS
INPUT TRANSITION MATRIX INCREMENT FOR EACH DOF
ENTER NSDF REAL VALUES

(DELTAV(I) > Ø)

DELTAD (REAL)
DISPLACEMENT
INCREMENTS
INPUT TRANSITION MATRIX INCREMENT FOR EACH DOF
ENTER NSDF REAL VALUES

(DELTAD(I) > Ø)

ICON (INTEGER)
ROTOR CONTROLS
INPUT TYPE Ø = NULL, 1 = GENERAL

if Ø:

no rotor controls input

if 1:

MROT (INTEGER)
NUMBER OF ROTORS
ENTER 1 INTEGER VALUE

(Enter the number of rotors at which controls will be applied; $1 \leq$ MROT ≤ 4)

for each rotor:

IRi (INTEGER)
ROTOR NUMBER
ENTER 1 INTEGER

(Enter the rotor number of the ith rotor for which there will be control inputs)

AØi (REAL)
COLLECTIVE ANGLE
(RAD)
ENTER 1 REAL VALUE

(Rotor fundamental control inputs are of the form

$$\theta_\emptyset + \theta_1 \cos \Omega t + \theta_2 \sin \Omega t$$

where $\theta_\emptyset$ is the collective input, θ_1 and θ_2 are the cyclic input components, and Ωt is the azimuth of blade 1; enter $\theta_\emptyset$)

A1Ci (REAL)
COSINE ANGLE (RAD)
ENTER 1 REAL VALUE

(Enter θ_1)

A1Si (REAL)
SINE ANGLE (RAD)
ENTER 1 REAL VALUE

(Enter θ_2)

NHi (INTEGER)
HIGHER HARMONIC
COLLECTIVE HARMONIC, Ø = NONE
ENTER 1 INTEGER VALUE

(Rotor higher harmonic control inputs are of the form

$$\theta_3 \cos n\Omega t + \theta_4 \sin n\Omega t$$

where θ_3 and θ_4 are higher harmonic collective input components, Ωt is the azimuth of blade 1, and n is the harmonic; enter n)

ANCi (REAL)
COS ANGLE (RAD)
ENTER 1 REAL VALUE

(Enter θ_3)

ANSi (REAL)
SIN ANGLE (RAD)
ENTER 1 REAL VALUE)

(Enter θ_4)

IPRINT (Y OR N)
PRINT EIGENVECTORS
ENTER 1 Y OR N VALUE

(Print out the roots of the Floquet transition matrix and the system eigenvectors [velocity, displacement])

3.3.6.2 Output, SSF3 - Output is the system eigenvalues and, optionally, the roots of the Floquet transition matrix and the system displacement and velocity eigenvectors. The system eigenvalues are related to the roots of the Floquet transiticn matrix by

$$e^{\lambda_{sys}} = \lambda_{floq}$$

and occur as conjugate pairs. The system eigenvalues are output as frequency (rad/sec) and damping (lb-sec/in., lb-sec-in./rad) pairs. A positive value for damping indicates an unstable system.

3.3.7 STR3 - Trim. STR3 combines periodic shooting (see paragraph 3.3.6) and the Newton-Raphson method, using an iterative scheme to find the periodic equilibrium state and control settings for a specified helicopter (single rotor) flight condition. Up to four independent direct controls and two indirect controls can be specified. The choice of direct controls depends on the rotor type and flight condition. The two indirect controls are two of the three Euler angles of the inertial (ground-fixed) coordinate system with respect to the body-fixed (nonrotating) system. Gravitational and centrifugal acceleration vectors can be specified in the inertial system. The acceleration vectors are automatically transformed to the body-fixed (global) coordinate system (see paragraph 3.4). If the global reference system is not used, the standard acceleration due to gravity is applied.

3.3.7.1 <u>Input, STR3</u> - The STR3 input prompts are shown below in capital letters. Only required input (based on previous input) is requested.

IFUS (Y OR N)
PERIODIC MOTION
OF FUSELAGE DOF TO BE CONSIDERED
ENTER 1 Y OR N VALUE

(STR3 ignores all system elastic degrees of freedom; fuselage rigid body degrees of freedom do not affect integration and the calculation of the initial conditions and control settings for each iteration, but are necessary for the unbalance force and moment calculations by which convergence to the trim condition (unbalance force and moments = Ø) is evaluated; if N is entered, the fuselage rigid body modes are omitted during integration (set to zero initial conditions), which is more efficient computationally and does not affect the result.

ITEET (Y OR N)
TEETERING ROTOR
ENTER 1 Y OR N VALUE

TRCOVA7 (REAL)
VALUE A
GIVE A, B, C VALUES SO THAT THE CYCLIC SINE CONTROL RELATES TO THE HORIZONTAL TAIL INCIDENCE IN THE FOLLOWING WAY:
INCIDENCE ANGLE = A + B * A1S + (C * A1S)**2
ENTER 1 REAL VALUE

(This relationship is imposed in order to provide effective fuselage moments for improved control performance with teetering rotors; A1S is the cyclic sine control setting A sin ωt [see paragraph 3.3.4.1, Input, STH3 - COLLECTIVE ANGLE])

TRCOVA8 (REAL)
VALUE B
ENTER 1 REAL VALUE

TRCOVA9 (REAL)
VALUE C
ENTER 1 REAL VALUE

H (REAL)
INITIAL INCREMENT
(SEC)
ENTER 1 REAL VALUE

(Enter the initial integration increment; H $\geq$ Ø)

TPER (REAL)
INTEGRATION PERIOD
ENTER 1 REAL VALUE

(Time period [sec] for integration; enter the rotation period of the rotor)

HTD (REAL)
SEPARATE INCREMENT
TIME DEP COEFFS
ENTER 1 REAL VALUE

(Separate integration increments may be entered, one each for time-dependent system coefficients and time-dependent system forces; if either input is nonzero, no error check will be performed; if an input is negative, the associated time-dependent parameters will not be computed; if zero is input, a separate increment will not be used)

HF (REAL)
SEPARATE INCREMENT
FORCE COMPUTATION
ENTER 1 REAL VALUE

E (REAL)
ERROR CHECK VALUE
IF Ø, THEN CONSTANT INCREMENT USED
ENTER 1 REAL VALUE

(The value entered for the error check defines the bounds for the computed state vector after each integration; the integration increment is automatically halved or doubled until the specified accuracy is achieved; if zero is input, the error check is omitted, which is more efficient computationally but may cause inaccuracies or numerical instability)

ITRIM (INTEGER)
CASE NUMBER

CASE 1 PERFORM FORCE BALANCE IN XCG, ZCG, ALFX, ALFY TRIM PARAMETERS: AØ, A1C, A1S, PTCH

CASE 2 PERFORM FORCE BALANCE IN XCG, YCG, ZCG, ALFX, ALFY TRIM PARAMETERS: AØ, A1C, A1S, AND 2 EULER ANGLES (AROLL, APTCH, AYAW)

CASE 3 PERFORM FORCE BALANCE IN XCG, YCG, ZCG, ALFX, ALFY, ALFZ TRIM PARAMETERS: AØ, A1C, A1S, 2 EULER ANGLES, AND YAW MOMENT

CASE 4 PERFORM FORCE BALANCE IN XCG, YCG, ZCG, ALFX, ALFY, ALFZ TRIM PARAMETERS: AØ, A1C, A1S, 2 EULER ANGLES, AND VEL VEHICLE

ENTER 1 INTEGER VALUE

NALLOW (INTEGER)
NO. OF ITERATIONS
ALLOWED
ENTER 1 INTEGER VALUE

(Enter the number of iterations after which the solution is to be terminated; 1 ≤ NALLOW ≤ 50)

CEA (Y OR N)
CONSTANT ERROR
ALLOWED
ENTER 1 Y OR N VALUE

if YES:

Enter the value for the error allowed in the elements of the state vector and in the control settings; the same value is applied to the velocity and displacement of all system degrees of freedom; the allowed error establishes bounds for the determination of the periodic equilibrium state.

if NO:

Enter the allowed error for the velocity and displacement of each system degree of freedom and for the control settings.

CEALLO (REAL)
CONSTANT ERROR
ALLOWED
ENTER 1 REAL VALUE

(Ø < CEALLO ≤ .0001)

EALLOWV (REAL)
VELOCITY ERRORS
INPUT ERROR FOR EACH DOF
ENTER NSDF REAL VALUES

(Ø < EALLOWV(I) ≤ .0001)

EALLOWD (REAL)
DISPLACEMENT ERRORS
INPUT ERROR FOR EACH DOF
ENTER NSDF REAL VALUES

(Ø < EALLOWD(I) ≤ .0001)

EALLOWT1 (REAL)
TRIM ERRORS ALLOWED
UNBALANCE FORCES AND MOMENTS CASE 1
ENTER 4 REAL VALUES

(Ø < EALLOWT1(I) ≤ .0001)

EALLOWT2 (REAL)
TRIM ERRORS ALLOWED
UNBALANCE FORCES AND MOMENTS CASE 2
ENTER 5 REAL VALUES

(Ø < EALLOWT2(I) $\leq$ .0001)

EALLOWT3 (REAL)
TRIM ERRORS ALLOWED
UNBALANCE FORCES AND MOMENTS CASE 3/4
ENTER 6 REAL VALUES

(Ø < EALLOWT3(I) $\leq$ .0001)

CI (Y OR N)
CONSTANT INCREMENT
FOR COMPUTING TRANSITION MATRIX
ENTER 1 Y OR N VALUE

if YES:

Enter the increment to be applied to the state vector and control settings for the perturbation of initial conditions (the periodic equilibrium state); the same increment is used for the velocity and displacement of all system degrees of freedom and for the control settings.

if NO:

Enter the increment for the velocity and displacement of each system degree of freedom and for the control settings.

CICRE (REAL)
CONSTANT INCREMENT
TO BE USED
ENTER 1 REAL VALUE

(CICRE > Ø)

DELTAV (REAL)
VELOCITY INCREMENTS
INPUT TRANSITION MATRIX INCREMENT FOR EACH DOF
ENTER NSDF REAL VALUES

(DELTAV(I) > Ø)

DELTAD (REAL)
DISPLACEMENT
INCREMENTS
INPUT TRANSITION MATRIX INCREMENT FOR EACH DOF
ENTER NSDF REAL VALUES

(DELTAD(I) > Ø)

DELTAT1 (REAL)
CONTROL VAR
INCREMENTS
INPUT TRANSITION MATRIX INCREMENT FOR EACH VARIABLE
ENTER 4 REAL VALUES

DELTAT2 (REAL)
CONTROL VAR
INCREMENTS
INPUT TRANSITION MATRIX INCREMENT FOR EACH VARIABLE
ENTER 5 REAL VALUES

DELTAT34 (REAL)
CONTROL VAR
INCREMENTS
INPUT TRANSITION MATRIX INCREMENT FOR EACH VARIABLE
ENTER 6 REAL VALUES

IWIND (Y OR N)
WIND VELOCITY
ENTER 1 Y OR N VALUE

(The wind velocities in all model force modules must be zero when STR3 is used; IWIND is not an option with Case 4)

VWINDG (REAL)
WIND VELOCITY
IN TERMS OF GROUND FIXED COORDINATES X, Y, Z
ENTER 3 REAL VALUES

(Enter the wind velocity components [in./sec])

VTRANG (REAL)
FUSELAGE TRANS VEL
TRANSLATIONAL VELOCITY IN TERMS OF GROUND FIXED COORD X, Y, Z
ENTER 3 REAL VALUES

(Enter the fuselage translational velocity components [in./sec]; VTRANG is not used with Case 4)

IROTIN (Y OR N)
FUSELAGE ANGULAR VEL
ENTER 1 Y OR N VALUE

(IROTIN is not an option with Case 4. Enter the fuselage angular velocity components [rad/sec])

VROTG (REAL)
FUSELAGE ANGULAR VEL
ANGULAR VEL IN TERMS OF GROUND FIXED COORD X, Y, Z

(Enter the fuselage angular velocity components [rad/sec])

CT (REAL)
ROTOR THRUST
ESTIMATE
ENTER 1 REAL VALUE

(Enter the estimated rotor thrust [lb] for the trim condition)

IGUESS (Y OR N)
INITIAL GUESSES
FOR CONTROLS
ENTER 1 Y OR N VALUE

(Enter Y if initial guesses for the velocity and displacement of the rotating system degrees of freedom and for the control settings are to be input)

TESTV (REAL)
INITIAL VELOCITY
ENTER NSDF REAL VALUES

(Enter guesses for rotating system DOFS, zero for all others)

TESTD (REAL)
INITIAL DISPLACEMENT
ENTER NSDF REAL VALUES

(Enter guesses for rotating system DOFS, zero for all others)

AØ (REAL)
COLLECTIVE COMPONENT
OF CONTROLS
ENTER 1 REAL VALUE

(See paragraph 3.3.4.1, Input STH3 - COLLECTIVE ANGLE)

A1C (REAL)
COSINE COMPONENT OF
CONTROLS
ENTER 1 REAL VALUE

A1S (REAL)
SINE COMPONENT OF
CONTROLS
ENTER 1 REAL VALUE

APTCH (REAL)
GUESS FUSELAGE ANGLE
APTCH
ENTER 1 REAL VALUE

(Enter initial guess for fuselage pitch angle [rad]; Case 1)

ICONST (INTEGER)
CONST EULER ANGLE
THERE ARE 3 EULER ANGLES FOR THE FUSELAGE WRT GROUND-FIXED COORDS
ENTER THE ONE YOU WANT TO KEEP CONSTANT - ROLL, PTCH, OR YAW

(ICONST is not an option with Case 1)

EULANGLE (REAL)
GUESS 3 EULER ANGLES
NOTE THE ONE YOU WANT TO KEEP CONSTANT WILL NOT BE CHANGED
ENTER 3 REAL VALUES

(radians; EULANGLE is not an option with Case 1)

YAWMON (REAL)
GUESS YAW MOMENT
YOU WANT TO ADD TO FUSELAGE TO ACHIEVE EQUILIBRIUM

(lb-in.; Case 3)

AAUTO (REAL)
DESCENDING ANGLE
OF AUTOROTATION

(radians; Case 4)

VAUTO (REAL)
AUTOROTATIONAL VEL
GUESS

(in./sec; Case 4)

ISTEAD (Y OR N)
STEADY TIME HISTORY
TO BE OUTPUT AFTER TRIM SOLUTION
ENTER 1 Y OR N VALUE

(If elected, time history solution will be initiated using initial conditions from last trim iteration)

CRT (Y OR N)
OUTPUT (at) THIS TERMINAL
ENTER 1 Y OR N VALUE

OUTTH (INTEGER)
OUTPUT DEVICE
NO. FOR TIME HISTORY SOLUTION
ENTER 1 INTEGER VALUE

ILOP (Y OR N)
SAVE STATE VECTORS
FOR INTERFACE, INTERNAL LOADS CALCULATIONS
ENTER 1 Y OR N VALUE

(Enter Y if time history state vectors are to be saved for loads calculations; see paragraphs 2.1 and 3.3.9)

JIIL (INTEGER)
INPUT I, EVERY ITH
STATE VECTOR TO BE WRITTEN TO LOADS FILE

($1 \leq$ JIIL ≤ 10)

GS (REAL)
VERT ACCELERATION
(G/GØ)
ENTER 1 REAL VALUE

(Enter the vertical acceleration in the inertial system in g's; else zero if global reference system not used; see paragraph 3.4)

ICF (Y OR N)
CONSIDER CENTRIFUGAL
ACCELERATION
ENTER 1 Y OR N VALUE

(Enter Y if centrifual acceleration due to steady turn is to be considered, else N; see paragraph 3.4)

(Note: turn rate is included in angular velocity vector, VROTG)

RT (REAL)
RADIUS OF TURN
(FT)
ENTER 1 REAL VALUE

(RT $\geq$ Ø)

3.3.7.2 Output, STR3 - For each iteration, the approximated initial conditions of system degrees of freedom, controls, global transformation matrix (optional), aerodynamic forces and moments, and unbalanced forces and moments in all of the trim directions are output. Units are lb, in., and rad.

INITIAL CONDITIONS

(DOF velocity and displacement; collective and cyclic controls; Euler angles)

INERTIAL TO GLOBAL TRANSFORMATION MATRIX

(Output for use with time history solutions; the rows of the matrix are the vectors of the global coordinate system in terms of the inertial coordinate system)

AERODYNAMIC FORCES

(Rotor forces and moments with respect to the hub; rotor torque and horsepower)

INDUCED VELOCITY, VELOCITIES WRT ROTOR AND FUSELAGE SYSTEMS

UNBALANCE FORCES

(Unbalance forces and moments with respect to the fuselage CG)

$$\left.\begin{array}{l}\text{FUSELAGE}\\ \text{WING}\\ \text{HTAIL}\\ \text{VTAIL}\end{array}\right\}\ \text{LIFT, DRAG, MOMENT, FX, FY, FZ}$$

3.3.8 SFD1 - Frequency Domain, Mobility. SFD1 computes the complex displacement mobility for the constant M, C, and K matrices of a system at specified frequencies. From the system mobility matrix, the user can obtain the response of a system or component degree of freedom due to a unit force at some system degree of freedom. The user selects component and system response degrees of freedom and selects the forced degrees of freedom from the system degrees of freedom. The user has the option to have output expressed as either displacement/unit force (in./lb, rad/in.-lb) or acceleration/unit force (g/lb).

The latter option can only be used with systems of strictly translational degrees of freedom. In addition, the output mobilities can optionally be written to an attached plot file which can be post processed for plotting or other purposes (see paragraph 2.1).

3.3.8.1 Input, SFD1 - The SFD1 input prompts are shown below in capital letters. Only required input (based on previous input) is requested.

FØ (REAL)
STARTING FREQ
ENTER 1 REAL VALUE

(The range of frequencies over which the mobilities will be calculated is defined by a starting frequency and an ending frequency; the discrete frequencies at which the mobilities will be calculated are determined by the frequency increment; enter the starting frequency [Hz]; FØ ≥ Ø)

FE (REAL)
ENDING FREQ
ENTER 1 REAL VALUE

(Hz; FE ≥ FØ)

FD (REAL)
INCREMENTAL FREQ
ENTER 1 REAL VALUE

(Hz; FD ≥ Ø)

IOU (INTEGER)
OUTPUT UNITS TYPE
1 = DISPLACEMENT/UNIT FORCE
2 = ACCELERATION/UNIT FORCE (G/LB)
ENTER 1 INTEGER VALUE

IPL (Y OR N)
WRITE PLOT FILES
ENTER 1 Y OR N VALUE

(See paragraphs 2.1 and 3.3.8)

RDOF (MODEL DOFS CHOSEN)
RESPONSE DOF

(The indexed system degree of freedom names are listed)

ALL SYSTEM DOFS (Y OR N)

(Enter Y if all system degrees of freedom are to be response degrees of freedom, else N)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the indices of the system response degrees of freedom, end with Ø)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the indices of the system response degrees of freedom, end with Ø)

ANY COMPONENT DOFS (Y OR N)

(Enter Y if there are to be any component response degrees of freedom, else N; if Y, the indexed component data set/data member names are listed)

SELECT COMPONENTS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the component indices, end with Ø)

For each component:

DOFS FOR COMPONENT ds/dm

(The indexed component degree of freedom names are listed)

SELECT DOF INDICES (Y OR N) OR QUIT COMPONENTS (Q)

(Enter Y if any degrees of freedom will be selected from this component, else N, continue with next component, else Q, terminate all component degree of freedom selection)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the indices of the component response degrees of freedom, end with Ø)

SDOF (SYSTEM DOFS CHOSEN)
FORCED DOF

(The indexed system degree of freedom names are listed)

ALL SYSTEM DOFS (Y OR N)

(Enter Y if all system degrees of freedom are to be forced degrees of freedom, else N)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter the indices of the system forced degrees of freedom, end with Ø)

3.3.8.2 Output, SFD1 - The real and imaginary parts of the complex displacement mobilities for the forced and response degrees of freedom are printed for each frequency (no imaginary part if no damping).

3.3.9 SII3 - Component Interface and Internal Loads. SII3 reads the system state vectors saved from a time history solution (STH3, STH4, STR3) and computes the interface loads (residual forces and moments) acting on component degrees of freedom. The residual force acting on a component is defined as

$$M\ddot{X} + C\dot{X} + KX - F$$

where M, C, and K are the component coefficient matrices, F is the component force vector, and X is the component state vector. The component state vector

is derived from the system state vector, and component coefficients are obtained by calling the component Coefficient and Active Modules (C---C, C---A). The time history interface loads can optionally be written to an attached plot file which can be post processed for plotting or other purposes (see paragraph 2.1).

The user also has the option to compute the time history internal loads for certain components. When this option is elected, the component Loads Module, C---L, is called and calculates the internal forces acting on component degrees of freedom. Internal forces are the component stiffness and damping forces which act on a degree of freedom with a given displacement and velocity. The internal force and ancillary calculations performed by the currently installed Loads Modules follow.

3.3.9.1 CSF1L - The forces exerted by the component springs and dampers are calculated. In addition, the associated strain energies and viscous damping energy dissipation rates are calculated.

Force

$$K_{ij} (X_j - X_i) + C_{ij} (\dot{X}_j - \dot{X}_i)$$

where K_{ij} and C_{ij} are elements of the component stiffness and damping matrices which multiply the differences in displacements and velocities of the associated component degrees of freedom.

Strain Energy

$$1/2\ K_{ij} (X_j - X_i)^2$$

Energy Dissipation Rate

$$1/2\ C_{ij} (\dot{X}_j - \dot{X}_i)^2$$

3.3.9.2 CES1L - The forces exerted by the spring-damper elements are calculated. In addition, the associated strain energies and viscous damping energy dissipation rates are calculated.

Force

$$K\ (X - \Delta - X_B) + C\ (\dot{X} - \dot{X}_B)$$

where K and C are the upper/lower stiffness and damping coefficients, X and $\dot{X}$ are the absolute displacement and velocity, Δ is the upper/lower gap, and X_B, $\dot{X}_B$ are the displacement and velocity of the base degree of freedom.

Strain Energy

$$1/2\ K\ (X - \Delta - X_B)^2$$

Energy Dissipation Rate

$$1/2\ C\ (\dot{X} - \dot{X}_B)^2$$

3.3.9.3 CRE3L, CRD3L - The in-plane bending moments, out-of-plane bending moments, and twisting moments due to shear are calculated at blade stations selected by the user. This option, the types of moments to be computed, and the blade stations at which they are to be computed must also be selected during input of the CRE3 or CRD3 data sets.

3.3.9.4 CRE3L, CRD3L Calculations

IN-PLANE BENDING MOMENT

$$EI_Y \left[\sum_{i=1}^{nv} V_i'' \, Y_i \sin\left(\sum_{j=1}^{np} \phi_j \, q_j + \theta \right) - \sum_{i=1}^{nw} W_i'' \, Z_i \cos\left(\sum_{j=1}^{np} \phi_j \, q_j + \theta \right) \right]$$

OUT-OF-PLANE BENDING MOMENT

$$EI_Z \left[\sum_{i=1}^{nv} V_i'' \, Y_i \cos\left(\sum_{j=1}^{np} \phi_j \, q_j + \theta \right) - \sum_{i=1}^{nw} W_i'' \, Z_i \sin\left(\sum_{j=1}^{np} \phi_j \, q_j + \theta \right) \right]$$

TWISTING MOMENT DUE TO SHEAR

$$GJ \sum_{i=1}^{np} \phi_i' \, q_i$$

V_i'', W_i''	second derivative of ith in-plane, out-of-plane mode shape (modal curvature)
ϕ_i	ith torsion mode shape (modal displacement)
Y_i, Z_i, q_i	amplitude of ith in-plane, out-of-plane, torsion mode
θ	built in twist
nv, nw, np	number of in-plane, out-of-plane, torsion modes
EI_Y, EI_Z	in-plane, out-of-plane bending stiffness
GJ	torsional stiffness

3.3.9.5 <u>Input, SII3</u> - The SII3 input prompts are shown below in capital letters. Only required input (based on previous input) is requested.

IFL (Y OR N)
INTERFACE LOADS
OPTION
ENTER 1 Y OR N VALUE

IFLDF (COMPONENT DOFS)
INTERFACE DOF
SELECT ONLY ONE COMPONENT

(The indexed component data set/data member names are listed)

SELECT COMPONENTS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter one component index, end with Ø)

DOFS FOR COMPONENT ds/dm

(The indexed component degree of freedom names are listed)

SELECT DOF INDICES (Y OR N) OR QUIT COMPONENTS (Q)

(Enter Y if any degrees of freedom will be selected from this component, else N or Q, terminate selection)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES-TO TERMINATE ENTER Ø

(Enter the indices of the component degrees of freedom for which interface loads will be calculated, end with Ø)

IFLPL (Y OR N)
PLOT INTERFACE LOADS
ENTER 1 Y OR N VALUE

(See paragraphs 2.1 and 3.3.9)

INL (Y OR N)
INTERNAL LOADS OPTION
ENTER 1 Y OR N VALUE

INLDF (COMPONENT DOFS)
INTERNAL DOF
SELECT ONLY ONE COMPONENT

(The indexed component data set/data member names are listed)

SELECT COMPONENTS BY INDICES
ENTER UNIQUE INTEGER VALUES - TO TERMINATE ENTER Ø

(Enter one component index, end with Ø)

DOFS FOR COMPONENT ds/dm

(The indexed component degree of freedom names are listed)

SELECT DOF INDICES (Y OR N) OR QUIT COMPONENTS (Q)

(Enter Y if any degrees of freedom will be selected from this component, else N or Q, terminate selection)

SELECT DOFS BY INDICES
ENTER UNIQUE INTEGER VALUES-TO TERMINATE ENTER Ø

(Enter the indices of the component degrees of freedom for which internal loads will be calculated, end with Ø)

3.3.9.6 Output, SII3 - If opted for, the time history internal loads are output first and are labeled with the component ds/dm. Output varies from component to component (see User's Manual Supplement, Volume III, for examples). If opted for, the time history interface loads follow the internal loads and are also labeled with the component ds/dm. The component interface loads are printed for each time step using the same format as that used by the other time history solutions.

3.4 GLOBAL REFERENCE SYSTEM

As an option, a global reference system may be defined for a model, permitting the application of consistent gravitational force vectors to the components of a model during time history and trim solutions. Computation of centrifugal forces due to steady turns may also be elected. The gravitational and centrifugal acceleration vectors are specified in an inertial (fixed) coordinate system and are successively transformed to the global (model) coordinate system and component coordinate systems during a solution. Forces are computed as the products of the component masses and the tranformed accelerations.

3.4.1 Inertial Coordinate System. The gravitational acceleration vector is defined in the - Z_I direction of the inertial coordinate system shown in Figure 23. For the case of a steady turn, the coordinate system assumes rotation

about the Z_I axis with a constant velocity Ω. The centrifugal acceleration vector is defined in the Y_I direction at a distance R from the Z_I axis ($\Omega^2 R$).

3.4.2 Global Coordinate System. The global coordinate system (Figure 23) is the reference coordinate system for the model. The global system is fixed in the inertial system and rotates with the Y_I axis. The gravitational and centrifugal accelerations are transformed to the global coordinate system; no other acceleration terms are considered at this level. Accelerations within component rotating systems are treated independently. g, Ω, and R are input to the time history and trim solutions, where the inertial acceleration vector is formed.

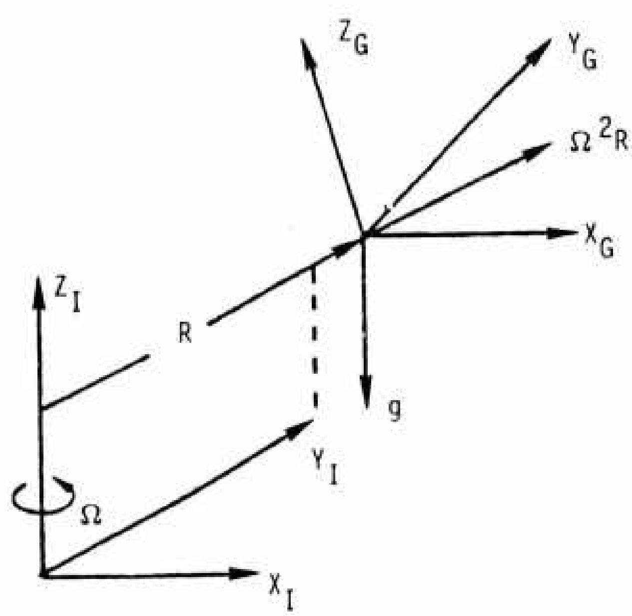


Figure 23. Inertial and Global Coordinate Systems.

3.4.2.1 Time History Solutions - For time history solutions, the user must provide the direction cosines of the global X and Y coordinate axes with respect to the inertial coordinate system, that is, the global X and Y vectors in terms of the inertial coordinates:

$$\vec{X}_G = a_x\vec{i} + a_y\vec{j} + a_z\vec{k}$$

$$\vec{Y}_G = b_x\vec{i} + b_y\vec{j} + b_z\vec{k}$$

where $\vec{i}$, $\vec{j}$, $\vec{k}$ are the unit vectors of the inertial coordinate system. The cross product of $\vec{X}_G$ and $\vec{Y}_G$ yields

$$\vec{Z}_G = C_x\vec{i} + C_y\vec{j} + C_z\vec{k}$$

Thus, the transformation of the inertial acceleration vector to the global coordinate system is given by

$$\left\{ a_G \right\} = \begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_z & c_y & c_z \end{bmatrix} \begin{Bmatrix} 0 \\ \Omega^2 R \\ -g \end{Bmatrix}$$

This transformation is performed in the time history solution Active module (STH3A, STH4A) and does not change during a particular solution; the global acceleration vector A_G remains constant.

3.4.2.2 Trim Solutions - In trim solutions, the Euler angles of the global coordinate system with respect to the inertial system are recomputed for each iteration. Hence, the transformation of the inertial acceleration vector to the global coordinate system must be repeated and is given by

$$\left\{ a_G \right\} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 0 \\ \Omega^2 R \\ -g \end{Bmatrix}$$

where ϕ, θ, ψ are the roll, pitch, and yaw angles of the global coordinate system with respect to the ground-fixed system for a given iteration.

3.4.3 Component Coordinate System. Body dimensions are assumed to be small with respect to the turn radius R; therefore, the origins of the component coordinate systems are defined to be coincident with the global origin. Also, the component systems are fixed in the global system.

The user must provide the direction cosines of the component X and Y coordinate axes with respect to the global coordinate system. The model sequence numbers and X and Y vectors for each component are input to CGLØ. The component transformation matrices are formulated in CLGØD and stored in the shared common block /GLOCOR/, from which they will be retrieved by the solution modules. Transformation of the global acceleration vector to the component coordinate systems immediately follows the computation of the global acceleration vector, and the resultant component acceleration vectors are stored in /GLOCOR/.

The forces due to the component acceleration vectors are applied to the rigid body translational degrees of freedom of the components in the component Active modules, C---A:

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = \begin{bmatrix} M & & \\ & M & \\ & & M \end{bmatrix} \begin{Bmatrix} A_{cx} \\ A_{cy} \\ A_{cz} \end{Bmatrix}$$

The component acceleration vectors a_c multiply the mass associated with the rigid body translational degrees of freedom. The resulting forces are then included in the component equations of motion. Because of the small body assumption, rigid body moments are neglected. The effect of acceleration on elastic modes is small and can also be neglected.

3.4.4 Local Coordinate System. Further transformation is required for components which define lumped parameter systems (CSF1, CSF3) and for rotor components (CRR2, CRE3, CRD3), for which rotor shaft angles and, hence, the orientation of the hub degrees of freedom can be defined with respect to the fuselage.

3.4.4.1 Lumped Parameter Systems - For the lumped parameter components, the user must provide the direction cosines of the local degree of freedom vectors with respect to the component coordinate system. The transformation matrix is formulated in the component coefficient module C---C and stored in the component private common block /C---/. The transformation matrix has a row dimension of 3 and a column dimension equal to the number of component degrees of freedom. Rows associated with rotational degrees of freedom are made null rows and the elements of rows associated with translational degrees of freedom are multiplied by the corresponding masses.

Forces are applied to the component degrees of freedom in the component Active modules C---A:

$$\underset{NDOFX1}{\left\{ F \right\}} = \underset{NDOFX3}{\left[H_{CL} \right]} \underset{3x1}{\begin{Bmatrix} a_{cx} \\ a_{cy} \\ a_{cz} \end{Bmatrix}}$$

F is the component force vector and H_{CL} is the transformation matrix (mass included).

3.4.4.2 Rotor Systems - When rotor shaft angles other than zero are specified for data sets of the fuselage component CFM2 and the global reference system option is elected, the fuselage and rotor component direction cosines given by the user for the CGLØ data must be identical. The local acceleration vector at the hub is computed in the rotor component Active module and is given by

$$\left\{ a_L \right\} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \delta_L & \sin \delta_L \\ 0 & -\sin \delta_L & \cos \delta_L \end{bmatrix} \begin{bmatrix} \cos \delta_F & 0 & -\sin \delta_F \\ 0 & 1 & 0 \\ \sin \delta_F & 0 & \cos \delta_F \end{bmatrix} \left\{ a_C \right\}$$

where δ_L is the lateral shaft angle and δ_F is the forward shaft angle. The forces due to local acceleration at the hub are then computed:

$$\begin{Bmatrix} F_{xHUB} \\ F_{yHUB} \\ F_{zHUB} \end{Bmatrix} = \begin{bmatrix} M & & \\ & M & \\ & & M \end{bmatrix} \begin{Bmatrix} a_{LX} \\ a_{LY} \\ a_{LZ} \end{Bmatrix}$$

M includes the total integrated blade mass and the mass of the hub.

4.0 ADDITION OF TECHNOLOGY MODULES

4.1 DEVELOPMENT OVERVIEW

4.1.1 Technology Module Library.

Technology Modules - The user is provided analytical design capabilities through the DYSCO library of Technology Modules. There are three types of Technology Modules which can be installed:

Component Technology Module	-	Represents an individual part of an analytical model.
Force Technology Module	-	Represents a set of applied force algorithms.
Solution Technology Module	-	Represents a particular solution method.

Naming Convention - Technology Modules have a four-character name; the first character is the alpha "C", "F", or "S", and denotes whether it is a component, force, or solution Technology Module. The second and third are general descriptors, and the fourth is an integer which indicates a general level of complexity. Examples are:

Components:	CRR2 - Rotor, rigid blades
	CSF1 - Structure, finite element
Forces:	FRAØ - Rotor aerodynamics, linear
	FSS1 - Sinusoidal shaker
Solutions:	STH3 - Time history
	SEA3 - Eigenanalysis
	SEA4 - Eigenanalysis

Technical Modules - Each Technology Module is implemented as a group of specially developed subprograms, each of which is called a Technical Module. The

type and the number of Technical Modules required for implementation depends upon the Technology Module type. Each Technical Module has specific functions to perform although the data required, the algorithms, and the output data varies. Thus, each has a unique "role" to play in the basic DYSCO modeling scenario. The relationship between this scenario and the Technical Modules is shown in Figure 24.

The name of a Technical Module is five characters; the first four are the name of the Technology Module and the fifth indicates the function to be performed. A brief description of those required are shown below, where C---, F---, S--- are the names of the Technology Modules:

Component Technical Modules

Input	C---I	Defines the user input data requirements
Definition	C---D	Defines the degrees of freedom and implicit relationships for a particular usage
Coefficient	C---C	Computes the constant coefficients and forces of the component equations
Active	C---A	Computes time varying coefficients and forces of the component equations based on component state vector. Accesses optional "force module" for computation of applied forces.
Block	C---B	Reads/writes private common block /C---/
Loads	C---L	Optional. Computes time history component loads based on component state vector. Accessed only by SII3A.

Force Technical Modules

Input	F---I	Defines the user input data requirements
Coefficient	F---C	Performs preliminary calculations

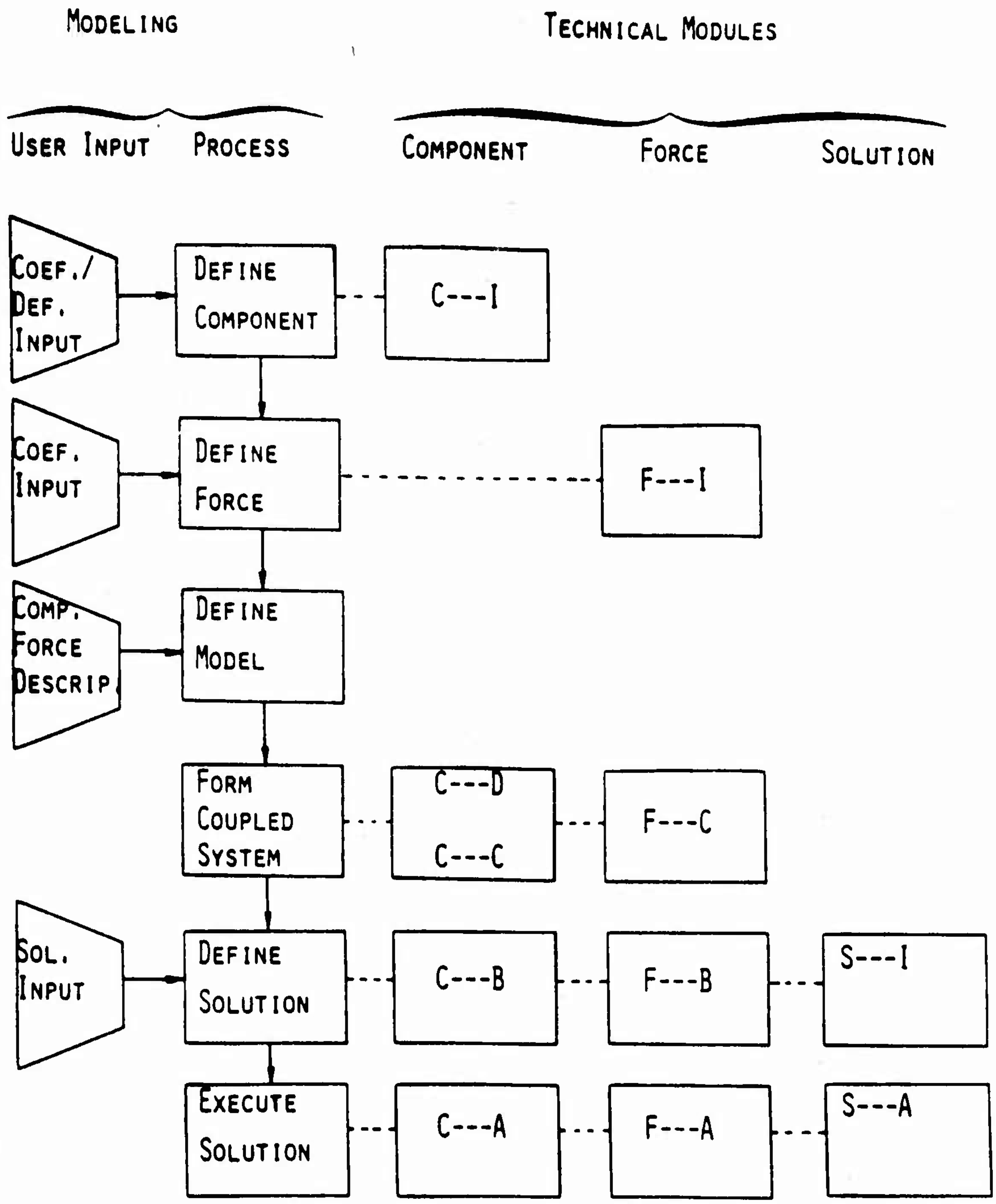


Figure 24. Relationship Between Modeling Scenario Technical Modules.

Active	F---A	Computes time varying applied forces of the associated component equation
Block	F---B	Reads/writes private common block /F---/

Solution Technical Modules

Input	S---I	Defines the user input data requirements
Active	S---A	Performs the solution and outputs the results

Communication - Various mechanisms are provided for communication between the DYSCO Executive and Technology Modules, between different Technology Modules, and within a specific Technology Module.

Between the DYSCO Executive and Technology Modules: The Executive is the "owner" of various data related to the model being formed or executed with a chosen solution method. Technical Modules (for a Technology Module) may require this data as input or may be required to compute this data and send it to the Executive for storage. These requirements are specific to the particular Technical Module being developed. The mechanisms utilized are as follows:

1. Argument Lists - Provide input from the Executive and output to the Executive.
2. XBG Utilities - Provide for retrieval of Base and Global Variables (user input for Component, Force, and Solution modules).
3. /AFTAB/ - A common block used by the Executive to store all airfoil tables required for a model. /AFTAB/ is incorporated into specific Technical Modules for accessing the tables.

Between Technology Modules: Communication between the Technology Modules is via special "shared" common blocks. The name of a shared

common block is descriptive of its usage. Data can be stored and retrieved by various Technical Modules as shown in Figure 25.

Within a Technology Module: Communication between the Technical Modules for a Component or Force Technology Module is performed through a "private" common block. The name of this common block is identical to the Component or Force for which it provides communication. The general flow of communication is shown in Figure 26.

Installation - Whenever a new Technology Module is added to the DYSCO library, certain information must be provided to the Executive. This information is provided by "installing" the Technology Module into the common block /XTM/, which contains master variables and arrays describing the limitations and usages of the Technology Module. Additionally, FORTRAN "calls" are inserted into specific predeveloped subprograms for each of the Technical Modules developed.

A new shared common block is similarly installed into DYSCO via the common block /XTM/ and insertion of appropriate "calls."

4.1.2 DYSCO Input Processor (DIP). The DIP provides for the creation, the maintenance, and the retrieval of input data supplied by the user for a specific usage of a Technology Module. It provides for consistency in prompting the user for input, in validating the input, and in handling erroneous input. It also directs editing of Component and Force input data at the individual item level. DIP performs these tasks by utilizing "knowledge tables," which are built specifically for each Technology Module to describe its input requirements.

User input for a Technology Module is of two types: Base and Global Variables. Component and Force Base Variables are viewed as "local" to those Technology Modules and are prompted for by DIP and saved as a ds/C--- or ds/F---. Similarly, Solution Base Variables are prompted for by DIP when the solution is run. The Global Variables are viewed as belonging to a model;

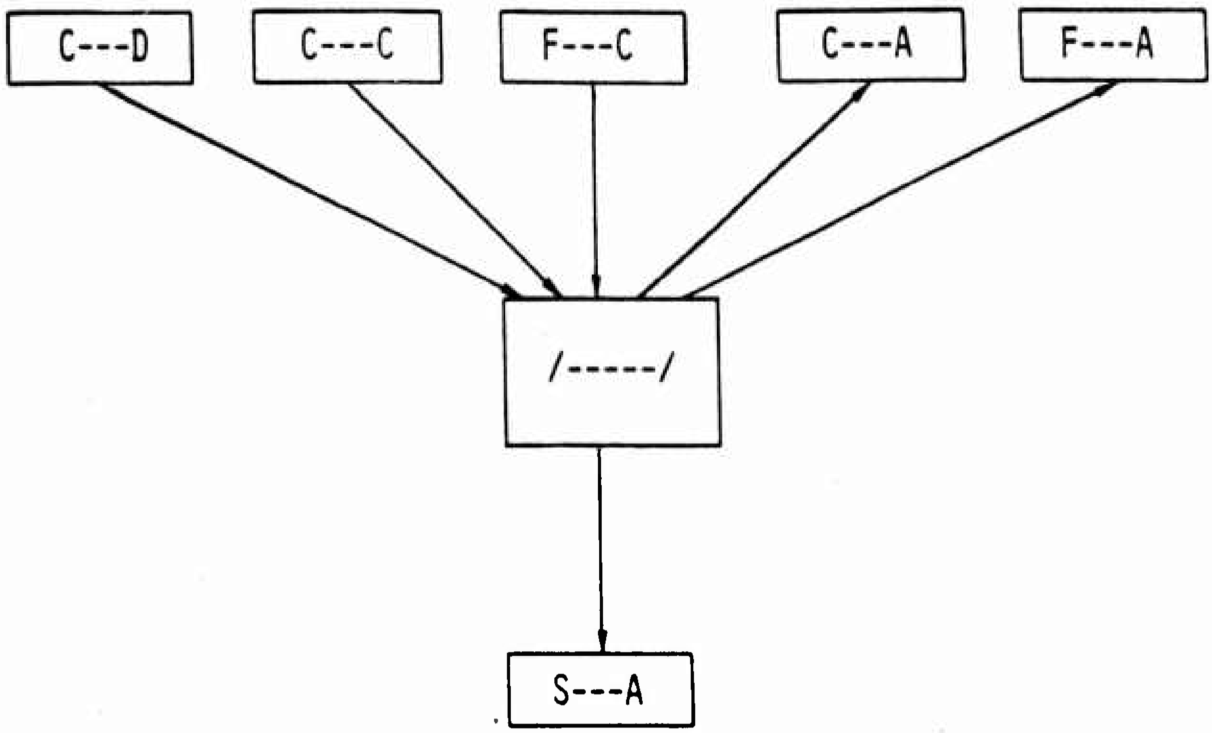


Figure 25. Shared Common Block.

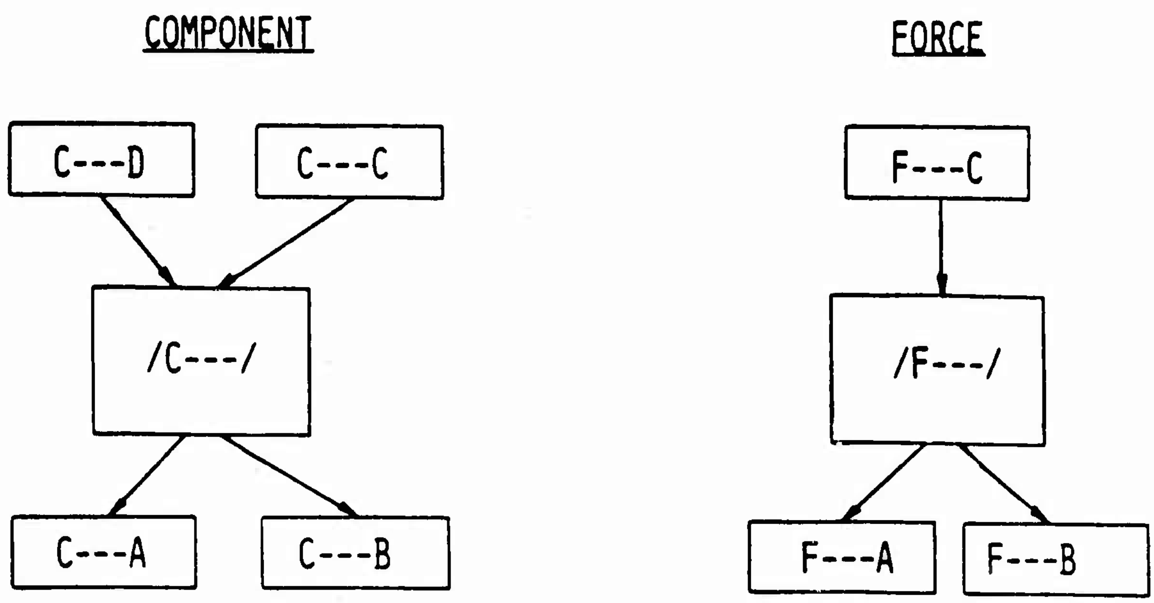


Figure 26. Private Common Blocks.

they are defined as a requirement for a Component or Force, but may also be required by other Components and Forces. DIP forms a unique set of the Global Variables required for all Components and Forces of a model, prompts the user for the values, and saves these values as part of the ds/MODEL. The user-supplied data is retrieved during the Definition and Coefficient phase of system coupling by the C---C, C---D, and F---C modules and during a solution by the S---A module, via the XBG utilities.

Additionally, a set of constants defined at the Executive system level are available for use in defining Base and Global Variables and also are available for Definition and Coefficient phase retrieval. These are called Global Constants. Finally, there is a set of values unique to a given model which can be used in defining Base Variables for a Solution. These are called Model Variables.

4.1.3 Executive Common Blocks. Several Executive common blocks are relevant to the development of a Technology Module:

/AFTAB/	-	Storage of all unique airfoil tables (ds/AIRFOIL) required by a model (ds/MODEL)
/XBG/	-	Definitions of Global/Model Variables and Global Constants
/XMODEL/	-	Model and coupled system variables
/XTM/	-	Master installation arrays for Components, Forces, Solutions, and shared common blocks
/TCI/	-	Technical and convenience constants.

4.1.4 Utilities. Utilities are provided to support development of Technical Modules for the DYSCO system. These utilities are designed to streamline code development and provide consistency within the system as a whole. Technical Modules are required in some instances to use these utilities, such as in retrieving Base and Global Variable input, while in other instances they may be

used at programmer discretion. Since the Technical Modules are dependent upon these Executive utilities, they are generally described as highly integrated with the DYSCO system.

There are two types of utilities provided to aid in code development:

XBG--	-	Base/Global Variable retrieval utilities
others	-	General purpose utilities.

4.2 DYSCO INPUT PROCESSOR (DIP)

4.2.1 Overview. The DIP allows for the creation, the maintenance, and the retrieval of data required as input for a Technology Module. General needs which are addressed by DIP are as follows:

1. Provide consistency in prompting for all input data and in handling erroneous input data.
2. Allow editing of input data at the individual item level (Component, Force).
3. Provide for two categories of input data: data local to a Technology Module, and data which is globally required by more than one component/force in a model.
4. Reduce the programmer effort required to introduce a new Technology Module.

DIP is a collection of Executive modules which utilize "knowledge tables" to satisfy these needs. The input data is described to DIP via a knowledge table which contains information such as variable name, type, prompt message, number of elements, and range constraints. It also contains criteria which is used by DIP to decide whether or not the data is needed in a particular input sequence. If DIP decides a piece of data is not required, the user will not

be prompted for its values. A knowledge table describing the input data requirements for a particular component/force/solution is defined in the C---I/F---I/S---I module by the programmer.

A complete set of component or force input data which has been created by the user under the guidance of DIP can later be edited at the individual item level, again under the guidance of DIP, with the knowledge table providing the basis for intelligent editing of the original data. DIP maintains the integrity of all of the input data by (1) ensuring that range constraints continue to be satisfied even though a piece of data may not itself have changed, and (2) ensuring that values have been provided for all data items needed even though an item may not have been needed in the original data being edited.

The knowledge table is also used by DIP to retrieve the data values and make them accessible during coefficient and definition phases of the dynamic coupling process and during a solution. Values are retrieved via DIP utilities (XBG utilities in paragraph 4.7.1) which are callable from C---C/C---D/F---C/S---A technical modules. The programmer is responsible only for defining the tables in the C---I/F---I/S---I modules and for requesting values when needed for technical computations. The actual processes required for creation, maintenance, and retrieval are performed by DIP.

Terms which are important to programming new Technology Modules are as follows:

> Base Variables - A Base Variable is input data which is local to a component/force/solution. In the case of a component or force, it is used for a specific occurrence of the component/force and is not to be used by another occurrence of the same or different component/force. Component and Force Base Variables, or base data, are saved on a ds/C--- or ds/F---. Base Variables are defined in the C---I/F---I/S---I module via IBTAB.

Global Variables - A Global Variable is input data which may be required by more than one component/force in a model. These data are associated with a model and saved on a ds/MODEL. Descriptions of all Global Variables are defined by IGTAB in common block /XBG/.

Those which are needed by a particular component/force are specified in the C---I/F---I module via IGVTAB. The developer of a component/force Technology Module may add Global Variables to DYSCO as needed.

Model Variables - A Model Variable is data unique to a given model which is required by a solution. These data are restricted to variables in /XMODEL/ which are single integer values and whose value is dependent upon the model. Descriptions of all Model Variables are defined by IXMTAB in common block /XBG/. Model Variables may only be used in S---I modules.

Global Constants - A Global Constant is a value which does not change and which is used in conjunction with Base/Global Variables. A constant may be used to indicate software design limits, such as a maximum dimension used. It may also be used for technical constants, such as a unit conversion factor.

Existence Criteria - Conceptually, each Base/Global/Model Variable and each Global Constant has an associated condition(s) which must be satisfied in order to "exist" and thus have a value(s). This condition is an existence criterion (EC) and may be specified when a Base/Global Variable is defined in IBTAB/IGTAB. If the EC for a particular variable is not satisfied, the user will not be prompted for the value. If no EC is specified, the variable is automatically considered to be existing, thus requiring a value to be specified by the user. The existence of any Global Constant or Model Variable is automatic and thus, each will always have a value.

Range Constraint - A range constraint (RC) is a condition which must be satisfied in order for a value to be acceptable as input. Base/Global Variables can have one or more associated with their

value(s). When a user is prompted for the value to be assigned to a variable and the user supplies the value, DIP verifies that all the RCs associated with that variable are satisfied.

4.2.2 Construction Codes. There are various codes used in the construction and usage of Base/Global/Model Variables and Global Constants. These are described in the following paragraphs.

4.2.2.1 TCODE - Type Codes - For a variable or constant, the TCODE defines the type of value which is valid. These codes are as follows:

TCODE		DESCRIPTION	VALID IN TABLES
1	R	Real	IBTAB/IGTAB/IGCTAB
2	I	Integer	IBTAB/IGTAB/IGCTAB/ IXMTAB
3	A	Alpha (A4)	IBTAB/IGTAB/IGCTAB
4	Y	Yes or no. When read interactively, the user specifies "Y" or "N". Internally, the value is stored as either "YES" or "NO."	IBTAB/IGTAB/IGCTAB
5	D	Degree of freedom name (DOF) which is composed of two parts, a 4-character alpha part and a 4-digit integer part. The format is (A4, I4).	IBTAB/IGTAB
6	X	Airfoil table which is defined as ds/AIRFOIL. The value consists of a ds name (2A4) and a dm name of AIRFOIL (2A4).	IBTAB (F---I only)
7	S	Sequential file description (5A4).	IBTAB (F---I only)
8	MD	Model DOF. Unique degree of freedom name(s) selected from all system and component DOFs in a model.	IBTAB (S---I only)
9	SD	System DOF. Unique DOF(s) selected from system DOFs only.	IBTAB (S---I only)
10	CD	Component DOF. Unique DOF(s) selected from component DOFs in a model.	IBTAB (S---I only)

4.2.2.2 PCODE - Property Codes - The property code defines certain characteristics of the variable or constant. These include whether it is a single value, a vector (array of one dimension), or a matrix (array of two dimensions). For vectors, additional characteristics are also definable. The PCODEs and valid types are as follows:

	PCODE	DESCRIPTION	VALID TYPES
1	I	Vector with increasing elements, vector (i + 1) .GE. vector (i)	R, I
2	D	Vector with decreasing elements, vector (i + 1) .LE. vector (i)	R, I
3	U	Vector with unique elements; an array of DOF names, each of which is unique	D
4	V	Vector with no constraints	R, I, A, Y, D, MD, CD, SD
5	M	Matrix	R, I, Y
6	S	Single element	All types, except MD
7	IS	Vector with strictly increasing elements, vector (i + 1) .GT. vector (i)	R, I
8	DS	Vector with strictly decreasing elements, vector (i + 1) .LT. vector (i)	R, I

4.2.2.3 BCODE - Boolean Codes - The following Boolean connectors are used for defining existence criteria and range constraints:

BCODE	DESCRIPTION
AND	The logical "and" connector A AND B implies both A and B are true
OR	The logical inclusive "or" connector A OR B implies A or B or both are true.

BCODE Evaluation - A BCODE is used as a logical connector for multiple existence criteria or range constraints. Where a single EC or RC is viewed as a "condition," the general form of usage is:

condition-1 BCODE condition-2 BCODE ...condition-n

Parentheses, with leftmost nesting, are implied so that the general form is interpreted as:

((...(condition-1 BCODE condition-2)...) BCODE condition-n)

Evaluation is from right to left with condition-n evaluated first. Evaluation will cease whenever the truth or falsity of the entire statement can be determined. An example evaluation follows.

((condition-1 AND condition-2) OR condition-3)

Case 1. condition-3 is evaluated and found to be true. The entire statement will be true regardless of the evaluation of the other two conditions. Evaluation ceases and the entire statement is considered true.

Case 2. condition-3 is found to be false; thus, further evaluation is required. condition-2 is found to be false and thus, the entire statement is considered false.

Case 3. condition-3 is false and condition-2 is true; thus, further evaluation is required. If condition-1 is true, the entire statement will be true. If condition-1 is false, the entire statement will be false.

4.2.2.4 CCODE - Condition Codes - Condition codes are used in defining existence criteria (EC) and range constraints (RC). Their usage varies between EC and RC definitions, but the basic form is as follows

var1 condition-code var2

where

1. var1 and var2 are the same type (TCODE)

2. var2 is a single element (PCODE is S)

3. var1 TCODE and PCODE depend on condition code.

Table 2 gives the CCODEs and the TCODE and PCODE combinations which are valid for var1.

4.2.3 <u>Base/Global/Model Variables Table Construction (IBTAB/IGTAB/IXMTAB)</u>. The construction of tables to define either Base, Global, or Model Variables is the same in concept. The structure of the three types of tables is identical; however, there are minor variances in content.

A Base Variables table, IBTAB, provides a description of the input data required for a particular Technology Module. IBTAB is defined via a sequence of DATA statements in the C---I/F---I/S---I Technical Module.

A Global Variables table, IGTAB, provides a description of the input data which may be required for more than one component or force. It is defined via a sequence of DATA statements in common block /XBG/.

A Model Variables table, IXMTAB, provides a description of the model data which may be required by a solution input table. It is defined via a sequence of DATA statements in common block /XBG/.

The IBTAB/IGTAB/IXMTAB table is composed of two parts, a preface and a body. The preface contains dimensioning and usage information; the body contains an entry for each Variable required. IBTAB/IGTAB/IXMTAB is a FORTRAN INTEGER array of one dimension.

TABLE 2. CODES VALID FOR VAR1

CONDITION CODE	DESCRIPTION	VALID TCODE	VALID PCODE (for var1)
LE	Less than or equal to	R, I	All
LT	Less than	R, I	All
GE	Greater than or equal to	R, I	All
GT	Greater than	R, I	All
EQ	Equal to:		
	for RC or EC	R, I	All
	for EC or RC	A, Y	S
NE	Not equal to:		
	for RC or EC	R, I	All
	for EC or RC	A, Y	S
EL	Last element in vector equal to var2	R, I	I, D, V
EF	First element in vector equal to var2	R,I	I, D, V
blank	Used in EC definition to mean var1 must exist. var2 is blank.	All	All

Preface

WORD	DESCRIPTION
(1)	1 (2 if IXMTAB)
(2-3)	'IBTA','B' or 'IGTA','B' or 'IXMT', 'AB'
(4)	Maximum dimension of IBTAB/IGTAB/IXMTAB
(5)	Actual dimension of IBTAB/IGTAB/IXMTAB
(6)	number of entries in body (.GE. 0)

Body

Entry 1 - entry for first Base/Global/Model Variable beginning at word 7

.

.

.

Entry n - entry for last Base/Global/Model Variable

Entry Format

WORD	FIELD	FORMAT	DESCRIPTION
(1-2)	vname	2A4	Variable name which is unique with respect to this table being constructed.
(3)	length	I	Number of words in entry. Not used by IXMTAB.
(4-8)	message	5A4	Variable description to be used as prompt message and for input summaries. The type, format of input, and number of values required should not be included. Not used by IXMTAB.
(9)	type	A1	Type code (TCODE).

WORD	FIELD	FORMAT	DESCRIPTION
(10)	property	A1	Property code (PCODE). Not used by IXMTAB.
(11)	aaaoueerr	I	Packed integer. Not used by IXMTAB.
	aaa		(000-999) indicating number of words in additional prompt message. These are printed starting on a new line according to (16A4).
	o		Override indicator. 1 - If input violates a range constraint, the user will be given the option to override the constraint. Valid only for single integer, real, alpha, or yes/no valued variable. 0 - The input must satisfy range constraints.
	u		"used flag". Always 1 for base variable table. Always 0 for global variable table.
	ee		Number of existence criteria (00-99).
	rr		Number of range constraints (00-99).
(12)	1st dimen	2A4	Optional Present only if variable is vector or matrix. Name of a variable whose value is the first dimension. This is the number of elements in a vector or the number of rows in a matrix. Must be the name of either (1) a variable in this table which precedes this entry, (2) a Global Constant which is a single integer value, or (3) a Model Variable.

WORD	FIELD	FORMAT	DESCRIPTION
(...)	2nd dimen	2A4	Optional Present only if variable is matrix. Contains the name of a variable whose value is the number of columns in matrix. Must be the name of a variable meeting the same 3 criteria as - for 1st dimension.
(...)	add msg	nA4	Optional Additional promptive message. Present only if aaa is not zero (n = aaa).
(...)	EC		Optional Start of existence criteria if ee .GT. 0. Each existence criteria is separated by a Boolean operator (BCODE). Existence Criteria 1 BCODE Existence Criteria 2 BCODE . . . Existence Criteria last
(...)	RC		Optional Start of range constraints if rr .GT. 0. Each range constraint is separated by a Boolean operator (BCODE).

WORD	FIELD	FORMAT	DESCRIPTION
(...)	RC (cont.)		Range Constraint 1 BCODE Range Constraint 2 BCODE . . . Range Constraint last

Existence Criteria Format -

WORD	FIELD	FORMAT	DESCRIPTION
(1-2)	var1	2A4	The name of either (1) a variable defined previously in this table for which a value may or may not exist, (2) a Global Constant, or (3) a Model Variable.
(3)	condition	A2	From CCODE
(4-5)	var2	2A4	The name of either (1), (2), or (3) as for var1.

Interpretation - An EC is interpreted as (var1 condition var2). In order for an EC to be true, the following must be true: (1) var1 and var2 must have an existing value. That is, the variable must have previously occurred in this table and have met its EC, and thus have a value, or the variable must be a Global Constant or Model Variable; (2) the condition must hold.

An alternate form may be used where condition code and var2 are blank. The EC will then be true if var1 has an existing value.

NOTES:

1. var1 and var2 may or may not have existing values when this EC is evaluated
2. var1 and var2 are the same type (TCODE)
3. var2, if not blank, must be a single value (PCODE = S)
4. Valid uses of CCODE in paragraph 4.2.2.4
5. BCODE evaluation process in paragraph 4.2.2.3.

Range Constraint Format -

WORD	FIELD	FORMAT	DESCRIPTION
(1)	condition	A2	From CCODE
(2-3)	var1	2A4	The name of either (1) a variable defined previously in this table for which a value must exist, (2) a Global Constant, or (3) a Model Variable.
(4-5)	var2	2A4	The name of either (1), (2), or (3) as for var1 or blanks.

Interpretation - Where vname is the Base/Global Variable being defined, an RC is interpreted as either:

1. If var2 is not blanks:

   (vname condition (var1 * var2))

2. If var2 is blanks:

   (vname condition var1)

NOTES:

1. var1 must have an existing value when this RC is evaluated or a fatal error will occur
2. If var2 is not blanks, then var2 must also have an existing value

3. var1 and var2 must be the same type (TCODE) as vname
4. var1 and var2 must be single valued (PCODE = S)
5. Valid uses of CCODE in paragraph 4.2.2.4
6. BCODE evaluation process in paragraph 4.2.2.3.

<u>FORTRAN 66 Implementation Note</u>

The goal is to have IBTAB/IGTAB/IXMTAB as a single dimension integer array. If the table is small enough, it can be initialized with a single DATA statement with continuation lines. However, due to the limitation of 19 continuation lines and the FORTRAN 66 restriction on using implied do-loops in a DATA statement, an awkward implementation scheme is required. This entails setting up dummy integer arrays, initializing each with a separate DATA statement, and using EQUIVALENCE to map the values into the appropriate locations of IBTAB/IGTAB/IXMTAB.

An example table implementation is shown below:

```
      INTEGER  IBTAB(65), IDUM1(50), IDUM2(15)
      EQUIVALENCE (IBTAB,IDUM1), (IBTAB(51),IDUM2)
      DATA IDUM1 / 1,'IBTA','B', 65,65,3,
     1   'RBM ','    ',11, 'RIGI','D BO','DY M','ODES','    ',
     2           'Y', 'S', 0000010000,
     3   'NR  ','    ',33,'NO. ','OF R','OTOR','S   ','    ',
     4         'I','S',006010102,
     5         'ADDI','TION','AL P','ROMP','T HE','RE',
     6         'RBM ',' ','EQ','YES',' ',
     7         'LE','I4','    ',' ',' ','AND',
     8         'GE','IO',' ',' ',' '/
      DATA IDUM2 /
     1    'NROT','    ',15,'ROTO','R NU','MBER','    ','    ',
     2         'I','V',000010000,
     3         'NR  ',' ',' ',' '/
```

Command VAL

The command VAL should be executed to validate any IBTAB/IGTAB construction before any run-time usage.

4.2.4 Global Variables Selection Table (IGVTAB). An IGVTAB table gives a list of selected Global Variables. AN IGVTAB is required in each C---I and F---I module to provide the list of Global Variables required as input. An IGVTAB is composed of a preface and a body. The preface provides dimensioning and usage information; the body contains an entry for each Global Variable to be selected.

Preface

WORD	DESCRIPTION
(1)	2
(2-3)	'IGVT','AB'
(4)	Dimension of IGVTAB
(5)	2
(6)	number of entries in body (.GE. 0)

Body

WORD	FORMAT	DESCRIPTION
(7-8)	2A4	name of first Global Variable
(9-10)	2A4	name of second Global Variable
.		
.		
.		
(....)	2A4	name of last Global Variable

NOTES:

1. Each entry must have a unique name with respect to the table being built
2. Command VAL should be used to validate table construction prior to its run-time usage.

Example IGVTAB

```
      INTEGER IGVTAB (10)
      DATA IGVTAB/ 2, 'IGVT','AB', 10, 2, 2,
     1     'RHO',' ',  'VSOU','ND'/
```

4.2.5 Global Constants Table (IGCTAB). The Global Constants table (IGCTAB) resides in common block /XBG/ and defines constants which are accessible for processing Base/Global Variables. It is composed of a preface and body. The preface provides dimensioning and usage information; the body contains an entry for each Global Constant to be defined.

Preface

WORD	DESCRIPTION
(1)	2
(2-3)	'IGCT','AB'
(4)	Dimension of IGCTAB
(5)	4
(6)	number of current entries

Body

WORD	DESCRIPTION
Entry 1	Entry for first Global Constant beginning at IGCTAB(7)
.	
.	
.	
Entry n	Entry for last Global Constant

Entry Format

WORD	FORMAT	DESCRIPTION
(1-2)	2A4	name of Global Constant with conventions as follows: X used as a suffix denotes some sort of maximum value is being defined (e.g., NROTX) Integers with no special meaning use forms such as I20, I-30 Real numbers with no special meaning use forms such as R90, R-90, R1.5, R-.50.
(3)	A1	Type code (valid TCODEs are R, I, A, Y).
(4)	-	Value with format depending on type.

NOTES:

1. Each entry must have a unique name with respect to IGCTAB
2. The full table in /XBG/ should be referenced for examples
3. When modifying this table, the command VAL should be used to validate table construction prior to run-time usage.

4.2.6 Run-Time Errors.

1. In general, the philosophy of the input processor is that any error which occurs at run-time during the actual usage of the table and which is due to negligence of the programmer will cause an error message to be printed and run termination. Thus, it is important to validate all new table construction via the VAL command, which will detect any violation of the specifications.

2. Overflow of Base/Global Variables values table (IBTMP/IGTMP) will be encountered if insufficient space has been allocated to store the values as the input processor receives them from the

user. The tables in which the actual values are stored are IBTMP1, IBTMP2 for Base Variables and IGTMP1, IGTMP2 for Global Variables; they are in /XBG/. An error may also occur at other points during the run; for example, if a previous DYSCO version allowed a large values table to be built and saved as a ds/C---, and a subsequent reduction in a table size resulted in insufficient space to restore the values. If errors of this nature occur, the tables in /XBG/ should be increased. This will then require a recompilation of all DYSCO routines which include /XBG/.

4.3 COMPONENT DEVELOPMENT

4.3.1 Common Block Usage. The following common blocks are of interest in Component development:

Private Common Block /C---/: Provides for communication between the modules C---C, C---D, C---L, and C---A for a specific Component being developed. In general, the C---C and the C---D modules will store data in the common block which is to be accessed later during the Active Module, C---A. Because of the DYSCO command RERUN, which starts a new solution execution for a model which has just completed running via command RUN, the integrity of this common block must be maintained throughout execution of the Active Module. Thus, any data which has been stored by C---C and C---D for use by the Active Module must remain intact and must not be changed so that a RERUN will execute properly. The C---B module reads/writes the common block. The only modules which may contain this common block are C---C, C---D, C---L, C---A, and C---B.

Shared Common Block: Used for communication between a Component(s), Force(s), and Solution(s) module(s). A new Component may utilize an existing shared common block or may implement a new one. A new common or any changes to an existing common must be performed with caution, and care must be taken to ensure that any new variable name

added does not currently exist as a local variable to any module which utilizes it.

/TCI/: Provides constants used in unit conversions, as well as other more general constants. It may be used for convenience and efficiency.

/XTM/: Can be modified only for installation of the Component. It must never be included in any C--- module.

/XBG/: Can be modified to incorporate new Global Constants/Variables. It must never be included in any C--- module.

4.3.2 Component Technical Module Development.

4.3.2.1 Component Input Module (C---I) -

Purpose - The Component Input Module, C---I, defines to the DYSCO Input Processor (DIP) the input data which the user must supply. There are two types of input data, Base and Global Variables, which require construction of the IBTAB and IGVTAB arrays, respectively. A Component cannot require airfoil tables or sequential files. The C---I module is used by DIP whenever information about the input requirements for the Component are needed.

SUBROUTINE Statement

SUBROUTINE C---I (ICODE, NAME, LUN, DS, ERR)

Input Arguments

ICODE, NAME, LUN, DS(2) - Integers. Usage varies.

Output Arguments

ERR - Logical.

NOTE: The type declarations are essential in order for FORTRAN to perform properly at a lower level!!!

Code Body

```
C***
C   PURPOSE - DESCRIBE INPUT DATA FOR COMPONENT C---
C
C   AUTHOR  -
C
C   SYSTEM  - IBM ---
C
C   REVISION HISTORY
C     4.0 INITIAL DEVELOPMENT
C
C***
      INTEGER DS(2), IBTAB (-), IGVTAB (-)
      LOGICAL ERR
      INTEGER ( additional integer arrays needed to implement IBTAB)
      EQUIVALENCE ( if needed to implement IBTAB)
      DATA IBTAB / ----
      DATA IGVTAB / ---
      CALL XBGX2 ( ICODE, LUN, DS, IBTAB, IGVTAB, NAME, ERR )
      RETURN
      END
```

NOTE: See paragraph 4.2.3 for IBTAB construction and paragraph 4.2.4 for IGVTAB construction.

4.3.2.2 Component Definition Module (C---D) -

Purpose - The Definition Module defines the degrees of freedom (DOF) and implicit relationships related to the particular use of the component representation. It retrieves user input data, which has been stored by DIP, as required via XBG--- utilities. Utilizing this data, calculations are performed to prepare data which will be used by the Coefficient Module, C---C, the Loads Module, C---L, and

the Active Module, C---A. The data prepared varies with the component and is stored in either the private common /C---/ or a shared common block.

SUBROUTINE Statement

```
SUBROUTINE C---D ( NCDFX, NCIDFX, NCCOX, IREF,  nuan,  NCDFA,
                   CDFLA, CDFIA,  NCIDF, CIDFL, CIDFI, NCCO,
                   CCO,   CCODF,  CIND,  OUT,   ERR )
```

Input Arguments

- NCDFX - Maximum number of component degrees of freedom (dof). Used for dimensioning CDFLA, CDFIA.
- NCIDFX - Maximum number of component implicit degrees of freedom. Used for dimensioning CIDFL, CIDFI, CIND.
- NCCOX - Maximum number of implicit coefficients of component. Used for dimensioning CCO, CCODF.
- IREF - Specifies that this is the IREFth call of this component in model (.GT. 0 and .LE. CMAXO) where CMAXO is the value installed in /XTM/ for this component.
- nuan - Structure, rotor, or control system number. Use name NSTR, NROT, or NCSYS. It is the user-assigned component number.
- OUT - Integer. Logical unit number for printer file. Should not need this.
- ERR - Logical. Error return flag. Set to false on entry.

Output Arguments

- NCDFA - Number of component degrees of freedom (.GE. 0).
- CDFLA,CDFIA - Integer (NCDFX). Arrays of literal and integer parts of implicit dof names.
- NCIDF - Number of component implicit degrees of freedom.
- CIDFL,CIDFI - Integer (NCIDFX). Arrays of literal and integer parts of implicit dof names.

NCCO - Number of implicit coefficients of component (.GE. 0).

CCO - Real (NCCOX). Array of implicit coefficients.

CCODF - Integer (NCCOX). Array of pointers to component dof corresponding to implicit coefficient. For the Ith implicit coefficient, let K = CCODF (I); then CDFLA (K) and CDFIA (K) is the corresponding component dof.

CIND - Integer (NCIDFX). Array of starting indices of coefficients in CCO,CCODF for each implicit dof.

ERR - Logical. Error flag. Set to true if error occurs.

Code Body

Follow the general prologue shown for C---I. The code body has no standardized form.

4.3.2.3 Component Coefficient Module (C---C) -

Purpose - The Component Coefficient Module, C---C, computes the constant coefficients and forces of the component equations for the coupled system. It retrieves user input data via the XBG utilities. Calculations are performed to prepare both constant mass, damping, and stiffness matrices, and also a constant force vector which will be used by the Active Module, C- -A.

SUBROUTINE Statement

```
SUBROUTINE C---C (NCDFX, IREF, nuan, NCDF, CM, ICHM, CC, ICHC,
                  CK,    ICHK, CF,   ICHF, ERR)
```

Input Arguments

NCDFX - Maximum number of component dof. Used in dimensioning.

IREF - Specifies that this is the IREFth call of this component in model (.GT. 0 and .LE. CMAXO installed in /XTM/).

nuan - Structure, rotor, or control system number. Use name NSTR, NROT, or NCSYS. It is the user-assigned component number.

NCDF - Number of component dofs (.GT. 0).

ERR - Logical. Error return flag. Set to false on entry.

Output Arguments

CM - Real (NCDFX,NCDFX) component constant mass matrix.

ICHM - Type of mass matrix indicator. Regardless of type, all values must be specified for the full matrix.
= 0 null (all elements are zero)
= 1 diagonal
= 2 symmetric
= 3 general

CC - Real (NCDFX,NCDFX) component constant damping matrix.

ICHC - Type of damping matrix indicator (0, 1, 2, 3).

CK - Real (NCDFX,NCDFX) component constant stiffness matrix.

ICHK - Type of stiffness matrix indicator (0, 1, 2, 3).

CF - Real (NCDFX) force vector.

ICHF - Type of force vector indicator
= 0 null
= 1 not null

ERR - Logical. Error flag. Set to true if error occurs.

Code Body

Follow general prologue shown for C---I. The code body has no standardized form.

4.3.2.4 Component Active Module (C---A) -

Purpose - The Component Active Module, C---A, computes the time varying coefficients and forces of the component equations based on the

component state vector at any point in time. The Active Module accesses an optional "force module" specified by the user for the computation of applied forces.

SUBROUTINE Statement

```
SUBROUTINE C---A (MAXC,  NCDFX, NSDFX,  NSCOX,  NDIMX,  ISEQ,   NDIM,
                  IREF,  nuan,  FORD,   FREF,   NCDFA,  NSDF,   TRAN,
                  SCO,   SCODF, TIME,   CV,     CVDOT,  CVDDOT, Y
                  DERY,  CM,    ICHM,   CC,     ICHC,   CK,     ICHK,
                  CF,    ICHF,  ITD,    IF      ... )
```

Input Arguments

MAXC	- Maximum number of components. Used for dimensioning.
NCDFX	- Maximum number of component dofs. Used for dimensioning.
NSDFX	- Maximum number of system dofs. Used for dimensioning.
NSCOX	- Maximum number of system implicit coefficients. Used for dimensioning.
NDIMX	- 2*NSDFX. Used for dimensioning.
ISEQ	- Component sequence number in model.
NDIM	- Number of dofs in first order formulation (2*NSDF).
IREF	- Specifies that this is the IREFth usage of this module in the model.
nuan	- Structure, rotor, or control system number. Name should be NSTR, NROT, or NCSYS. This is the user assigned number of component.
FORD	- Ordinal of force in installation arrays in /XTM/. If 0, then force not applicable.
FREF	- Integer specifies that this is the FREFth usage of the specified force module in the model.
NDCFA	- Number of dofs of this Component.
NSDF	- Number of system dofs.
TRAN	- Integer (NCDFX,MAXC). Transformation table.
SCO	- Real (NSCOX). System implicit coefficients.

SCODF - Integer (NSCOX). System dof numbers corresponding to SCO.

TIME - Time.

CV - Real (NCDFX). Component displacement vector.

CVDOT - Real (NCDFX). Component velocity vector

CVDDOT - Real (NCDFX). Component acceleration vector.

Y - Real (NDIMX). System state vector. All velocity, all displacement.

DERY - Real (NDIMX). Time derivative of Y.

ITD - If 1, then compute time dependent coefficients.

IF - If 1, then compute force. (FORD .NE. 0)

Output Arguments

CM - Real (NCDFX,NCDFX). Generalized mass matrix for coupled system.

ICHM - Type of mass matrix indicator. Regardless of type, all values must be specified for the full matrix.
= 0 null (all values zero)
= 1 diagonal
= 2 symmetric
= 3 general

CC - Real (NCDFX,NCDFX). Generalized damping matrix for coupled system.

ICHC - Type of damping matrix indicator (0, 1, 2, 3).

CK - Real (NCDFX,NCDFX). Generalized stiffness matrix for coupled system.

ICHK - Type of stiffness matrix indicator (0, 1, 2, 3).

CF - Real (NCDFX). Generalized force vector for coupled systems.

ICHF - Type of force vector indicator.
= 0 null
= 1 not null

Code Body

The prologue should be similar to that shown for C---I. The code body is not standardized; however, a skeletal example is shown below to show how Force modules may be invoked. Here, only Force FFAO or FFC2 is valid for use with the Component. Since the Executive has performed validations, upon entry to this C---A the value of FORD will be 0 (if IF = 0) or either 2 or 5 (if IF = 1). FORD will never have the value 1, 3, 4, or a number higher than 5. In /XTM/, the position of FFAO in the installation arrays is 2; FFC2 has position 5.

```
      (prologue)
      (declarations)
      .
      .
      .
      IF (IF .EQ. 0) GO TO 900
      GO TO (900, 20, 900, 900, 50), FORD
   20 CALL FFAOA ( IREF, FREF, IXCG, PTCHD, VEL, FDRAG )
      GO TO 100
   50 CALL FFC2A ( IREF, FREF, TRCOVA, VTRANF, VWINDF, FXCG,
     1             FYCG, FZCG, MXCG,   MYCG,   MZCG,   CG,
     2             ITEET )
  900 CONTINUE
```

4.3.2.5 Component Block Module (C---B) -

Purpose - The Block Module, C---B, must optionally write or read the private common block /C---/ to a ds/CASE.

SUBROUTINE Statement

SUBROUTINE C---B (ICODE, LUN, NREC)

Input Arguments

ICODE - Action to be taken indicator.
= 'W' then values to be written
= 'R' then values to be read

LUN - Integer code of file for writing/reading.

NREC - Number of the first record for writing/reading.

Output Arguments

NREC - The number of the record following the last record written/read.

Code Body

The prologue should be similar to that shown for C---I. The code is not standardized; however, a skeletal example follows:

```
      (prologue)
      (declarations)
      INCLUDE (C---)                       *get common /C---/ *
      DATA IR/'R'/
      IF (ICODE .EQ. IR) GO TO 200
      WRITE (LUN'NREC, 1000) ...
      NREC = NREC + number of records written
      .
      .
      .
      GO TO 900
  200 READ (LUN'NREC,1000)...
 1000 FORMAT(  )
      NREC = NREC + number of records written
  900 RETURN
      END
```

NOTES:

1. The ds/CASE can be quite large in its entirety; thus, utmost efficiency should be used in minimizing the amount of data written to the file. For example, only those parts of array which actually contain values should be written. If a matrix is entirely zero and a "flag" is available or can be constructed, then the flag should be written with the absence of matrix values; the zero values can be reconstructed during the read phase.

2. The record format for all DYSCO files is 80-character. If less than an 80-character is written, FORTRAN will blank-fill. FORTRAN will write multiple records as required. Economy of file space is important.

4.3.2.6 Component Loads Module (C---L) -

Purpose - The Component Loads Module, C---L, computes and prints out the time history internal loads acting on the component degrees of freedom. C---L is called by the Solution Module, SII3A, following a time history solution from which solution state vectors have been saved (see paragraph 3.3.9).

SUBROUTINE Statement -

```
SUBROUTINE C---L (CV, CVDOT, CVDDOT, AM, AC, AK, AF, IREF, NCDF,
                  NCDFX, JND, MDFL, MDFI, NDOF, OUT, X)
```

Input Arguments

CV	- Real (NCDFX). Component displacement vector.
CVDOT	- Real (NDCFX). Component velocity vector.
CVDDOT	- Real (NDCFX). Component acceleration vector.
AM	- Real (NDCFX, NDCFX). Component mass matrix.
AC	- Real (NDCFX, NDCFX). Component damping matrix.
AK	- Real (NDCFX, NDCFX). Component stiffness matrix.

AF	-	Real (NCDFX). Component force vector.
IREF	-	Specifier that this is the IREFth usage of this module in the model.
NCDF	-	Number of DOFs in this component.
NCDFX	-	Maximum number of component DOFs. Used for dimensioning.
JND	-	INTEGER (NDOF). Indices of component DOFs for which internal loads will be calculated.
MDFL	-	(NDOF). Literal part of loads DOFs.
MDFI	-	(NDOF). Integer part of loads DOFs.
NDOF	-	Number of loads DOFs.
OUT	-	Output device number.
X	-	Time.

Code Body

Follow general prologue shown for C---I. The code body has no standardized form.

4.3.3 Component Installation.

Common Block /XTM/ - A new Component must be described to the DYSCO Executive via master variables/arrays in /XTM/. Refer to paragraph 4.6.1.4 for descriptions. Those which must be changed are:

NOCOM, CNAME, CTYPE, CMAXO, CNVF, CVFN, CDES, CNSCB, CNASCB

FORTRAN Call Statements - The subroutines below require insertions to accommodate the new Technical Modules. Placement of calls is dependent on position of Component in CNAME array in /XTM/. This position is the ordinal of the Component.

XBGX1	-	calls C---I ( )
XXCD	-	calls C---D ( )
XXCC	-	calls C---C ( )

FCT - calls C---A ( )

XCWRLC - calls C---B ( )

SII3A - calls C---C ( ), C---A ( ), C---L ( )

Shared Common Block - If a new shared common block has been developed, then it must be installed as described in paragraph 4.6.2.1. The Components, Forces, or Solutions with whom it is to communicate must have their installation arrays (CNSCB, CNASCB and FNSCB, FNASCB and SNSCB, SNASCB) modified.

Command VAL - The command VAL should be executed to validate the construction of Base and Global Variables tables.

4.4 FORCE DEVELOPMENT

4.4.1 Common Block Usage. The following common blocks are of interest in Force development:

Private Common Block /F---/ - Provides for communication between the modules F---C and F---A for a specific Force being developed. In general, the F---C module will store data in the common block which is to be accessed later during the Active Module, F---A. Because of the DYSCO command RERUN, which starts a new solution execution for a model which has just completed running via command RUN, the integrity of this common block must be maintained throughout execution of the Active Module. Thus, any data which has been stored by F---C for use by the Active Module must remain intact and must not be changed so that a RERUN will execute properly. The F---B module reads/writes the common block for a CASE. The only modules which may contain this common block are F---C, F---A, and F---B.

Shared Common Block: Used for communication between a Component(s), Force(s), and Solution(s) module(s). A new Component may utilize an existing shared common block or may implement a new one. A new common or any changes to an existing common must be performed with caution, and care must be taken to ensure that any new variable name

added does not currently exist as a local variable to any module which utilizes it.

/TCI/: Provides constants used in unit conversions, as well as other more general constants. It may be used for convenience and efficiency in any module.

/XTM/: Can be modified only for installation of the Force. It must never be included in any F--- module.

/XBG/: Can be modified to incorporate new Global Constants/Variables. It must never be included in any F--- module.

/AFTAB/: Contains airfoil tables for the model. When a model is RUN, the Executive reads all tables required into /AFTAB/ and provides information for accessing the table data. Common /AFTAB/ can be included in F---C or F---A.

4.4.2 Force Technical Module Development.

4.4.2.1 Force Input Module (F---I) -

Purpose - The Force Input Module, F---I, defines to the DYSCO Input Processor (DIP) the input data which the user must supply. There are two types of input data, Base and Global Variables, which require construction of the IBTAB and IGVTAB arrays, respectively. The input requirements are in three forms: raw data values, airfoil tables (ds/AIRFOIL), and sequential files. All three types may be Base Variables. Only raw data values may be Global Variables. The F---I module is used by DIP whenever information about the input requirements for the Force are needed.

Airfoil Tables - An airfoil table exists in the form of a ds/AIRFOIL and is created by the user via the create (CRE) command. Each table required is assigned a unique name in the Base Variable table IBTAB. DIP will prompt the user for the ds name of the table using the description given in its IBTAB entry. Information required to access

data in the airfoil table is obtained later (when a model is RUN) by the F---C module. There is a limit to the number of unique airfoil tables required by a model. This limit is defined in /AFTAB/. There is no check by the Executive at the Force input level for the limit; Executive checks at RUN time.

<u>Sequential Files</u> - A sequential file contains data not created within the DYSCO environment; for example, an induced velocity table. For each file required, the coder must define a Base Variable name to be associated with it, as well as a description. Each file will be described to DIP by a separate entry in the Base Variable table IBTAB. DIP will prompt the user for the file unit number for each sequential file. The unit number can be obtained later by F---C.

<u>SUBROUTINE Statement</u>

SUBROUTINE F---I (ICODE, NAME, LUN, DS, ERR)

Input Arguments

ICODE, NAME, LUN, DS(2) - Integers. Usage varies, but is not relevant to code development.

Output Arguments

ERR - Logical

<u>NOTE</u>: The type declarations are essential in order for FORTRAN to perform properly at a lower level!!!

<u>Code Body</u>

```
C***
C     PURPOSE - DESCRIBE INPUT DATA FOR FORCE F---
C
C     AUTHOR  -
C
C     SYSTEM  - IBM --
```

```
C
C    REVISION HISTORY
C      4.0 INITIAL DEVELOPMENT
C***
      INTEGER DS(2), IBTAB (-), IGVTAB (-)
      LOGICAL ERR
      INTEGER (additional arrays needed to implement IBTAB)
      EQUIVALENCE (if needed to implement IBTAB)
      DATA IBTAB / ---
      DATA IGVTAB / ---
      CALL XBGX2 (ICODE, LUN, DS, IBTAB, IGVTAB, NAME, ERR )
      RETURN
      END
```

NOTE: See paragraph 4.2.3 for IBTAB construction and paragraph 4.2.4 for IGVTAB construction.

4.4.2.2 Force Coefficient Module (F---C) -

Purpose - The Force Coefficient Module, F---C, calculates the force coefficients to be associated with a Component for the coupled system equations of a model. These values are saved, typically in common block /F---/, for later access by the Active Module F---A.

Airfoil Tables - When the Coefficient Module is executed, the airfoil tables required have been read by the Executive and reside in common block /AFTAB/. To access the data, the type and index must be given. This information is obtained via the XBGXS utility. It should be stored, typically, in private common /F---/, for later use by the Active Module.

Sequential Files - The Coefficient Module can obtain the file unit number (assigned by the user) for any sequential file by calling the XBGSS utility. This unit number should typically be stored in private common /F---/for later use by the Active Module.

SUBROUTINE Statement

SUBROUTINE F---C (MAXC, NCDFX, FREF, nuan, NCOMP, ISEQ, NCDFA, CDFLA, CDFIA, ERR)

Input Arguments

- MAXC - Maximum number of components in model. Used in dimensioning.
- NCDFX - Maximum number of component dofs.
- FREF - Integer. Specifies that this is the FREFth usage of this force module in the model.
- nuan - Rotor, Structure, or control system number (user assigned). Variable should be NROT, NSTR, or NCSYS.
- NCOMP - Number of components in model.
- ISEQ - Sequence number of the component (in the model) associated with this usage of the force.
- NCDFA - Integer (MAXC). Number of dofs for each component in model.
- CDFLA - Integer (NCDFX,MAXC). Literal part of dof names for each component in model.
- CDFIA - Integer (NCDFX,MAXC). Integer part of dof names for each component in model.
- ERR - Logical. Set to false on entry.

Output Arguments

- ERR - Logical. Error flag. Set to true if error occurs.

Code Body

The prologue should be similar to that shown for F---I. The code body is not standardized.

4.4.2.3 Force Active Module (F---A) -

Purpose - The Force Active Module, F---A, computes the applied forces for a given time and given component state vector(s). In a model, a force is associated with a component; F---A is called by the associated Component's Active Module. This is the only type of Technical Module which is called by another Technical Module.

Airfoil Tables - If an airfoil table is used, its type and its index have been stored by F---C to enable access to the table data by F---A. Use this information to retrieve data from /AFTAB/. For example, if a table is type 2 with index stored in variable INDEX, then data access is CL2(-,-,INDEX), CD2(-,-,INDEX), and so on. See paragraph 4.6.1.1 for description of airfoil tables.

Sequential Files - If a sequential file is used, its integer file unit has been stored by F---C to enable access. These are not DYSCO files and usage is up to the coder.

SUBROUTINE Statement

SUBROUTINE F---A (-------)

Arguments are not standardized; however, groups of Force modules which are optionally accessed by a particular Component Active Module should have the same argument list.

Code Body

The prologue should be similar to that shown for F---I; the code body is not standardized.

4.4.2.4 Force Block Module (F---B) -

Purpose - The Force Block Module, F---B, must optionally write or read the private common block /F---/ to a ds/CASE.

SUBROUTINE Statement and Code Body

The specifications for the Force Block Module are identical to those for the Component Block Module. See paragraph 4.3.2.5 for a description.

4.4.3 Force Installation.

Common Block /XTM/ - A new Force must be described to the DYSCO Executive via master variables/arrays in /XTM/. Refer to paragraph 4.6.1.4 for descriptions. Those which must be changed are:

NOFOM, FNAME, FMAXO, FDES, FNSCB, FNASCB

For each component with which this Force may be associated, the following must be changed:

CNVF, CVFN

FORTRAN Call Statements - The below subroutines require insertions to accommodate the new Technical Modules. Placement of calls is dependent on position of Force in FNAME array in /XTM/. This position is the ordinal of the Force.

XBGX1	- calls F---I ( )
XXFC	- calls F---C ( )
XCWRLC	- calls F---B ( )
C---A	- For each Component Technology Module with which this Force may be associated or applied, the C---A module must allow for a call to the Active Module F---A.

Shared Common Block - If a new shared common block(s) has been developed, then it must be installed as described in paragraph 4.6.2.1. The Components, Forces, or Solutions with whom it is to communicate must have their installation arrays (CNSCB, CNASCB and FNSCB, FNASCB and SNSCB, SNASCB) modified.

Command VAL - The command VAL should be executed to validate input data table construction.

4.5 SOLUTION DEVELOPMENT

4.5.1 Common Block Usage. The following common blocks are of interest in Solution Development:

Shared Common Block: Used for communication between a Component(s), Force(s), and this Solution's modules. A new Solution may utilize an existing shared common block or may implement a new one. A new common or any changes to an existing common must be performed with caution, and care must be taken to ensure that any new variable name added does not currently exist as a local variable to any module which utilizes it.

/TCI/: Provides constants used in unit conversions, as well as other more general constants. It may be used for convenience and efficiency in any module.

/XTM/: Can be modified only for installation of the Solution. It must never be included in any S--- module.

/XBG/: Can be modified to incorporate new Model Variables and Global Constants. It must never be included in any S--- module.

4.5.2 Solution Technical Module Development.

4.5.2.1 Solution Input Module (S---I) -

Purpose - The Solution Input Module, S---I, defines to the DYSCO Input Processor (DIP) the input data which the user must supply. There is one type of input data, Base Variables, which requires construction of the IBTAB array. The S---I module is used by DIP whenever information about the input requirements for the Solution are needed.

SUBROUTINE Statement

SUBROUTINE S---I (ICODE, NAME, LUN, DW, ERR)

Input Arguments

ICODE, NAME, LUN, DS(2) - Integers. Usage varies.

Output Arguments

ERR - Logical.

NOTE: The type declarations are essential in order for FORTRAN to perform properly at a lower level!!!

Code Body

```
C***
C    PURPOSE - DESCRIBE INPUT DATA FOR COMPONENT S---
C
C    AUTHOR  -
C
C    SYSTEM  - IBM --
C
C    REVISION HISTORY
C      4.0 INITIAL DEVELOPMENT
C
C***
     INTEGER DS(2), IBTAB (-)
     LOGICAL ERR
     INTEGER (additional arrays needed to implement IBTAB)
     EQUIVALENCE (if needed to implement IBTAB)
     DATA IBTAB / ---
     CALL XBGX2 (ICODE, LUN, DS, IBTAB, IDUM, NAME, ERR)
     RETURN
     END
```

NOTE: See paragraph 4.2.3 for IBTAB construction.

4.5.2.2 Solution Active Module (S---A) -

Purpose - The Solution Active Module, S---A, performs the calculations for the solution method. It retrieves user input data via the XBG utilities.

SUBROUTINE Statement

SUBROUTINE S---A (...)

Input and output arguments are not standardized. The input may come from model and coupled system variables residing in common block /XMODEL/.

Code Body

The prologue should be similar to that shown for S---I; the code body is not standardized.

For solution methods which are relatively simple, such as Eigen or frequency domain solutions, executions of the active modules for the Components and Forces (C---A, F---A) are not required. However, for the more complex integration type of methods, their involvement is required. To develop such a solution requires more information than is presented in this document.

4.5.3 Solution Installation.

Common Block /XTM/ - A new Solution must be described to the DYSCO Executive via master variables/arrays in /XTM/. Refer to paragraph 4.6.1.4 for descriptions. Those which must be changed are:

NOSOM, SNAME, SDES, SNSCB, SNASCB

FORTRAN Call Statements - The below subroutines require insertions to accommodate the new Technical Modules. Placement of calls is dependent on position of Solution in SNAME array in /XTM/. This position is the ordinal of the Solution.

XBGX1 - calls S---I ( )
XXS - calls S---A ( )

Shared Common Block - If a new shared common block(s) has been developed, then it must be installed as described in paragraph 4.6.2.1. The Components and Forces with which it is to communicate, and other Solutions which may use it, must have their installation arrays (CNSCB, CNASCB and FNSCB, FNASCB and SNSCB, SNASCB) modified.

4.6 COMMON BLOCKS

4.6.1 Executive Common Blocks

4.6.1.1 /AFTAB/ -

Purpose - To provide for memory storage of airfoil tables (ds/AIRFOIL) required by forces in the current model. All unique tables are in memory when a model is running. A single table may be required more than once in a model; however, the table is stored only one time in memory. There is one type of table: type 2 (2 independent variables). The number of tables required by a model must not exceed MAXAF2. For accessing a specific table, the type and the index must be given.

Variables

AFDM - Integer (2) dm name for all airfoil tables 'AIRF', 'OIL '.

(Table Type 2 Variables)

MAXAF2 - Maximum number type 2 tables (5 for DYSCO 4.0).
NAF2 - Number type 2 tables required and currently in memory.
HEAD2 - Integer (20,MAXAF2) 80-character description of table.
NALFA2 - Integer (MAXAF2) number of angles of attack (.GT. 0 and .LE. 65).
NMACH2 - Integer (MAXAF2) number of Mach numbers NMACH2(index) (.GT. 0 and .LE. 9).

AMACH2 - Real (9,MAXAF2) Mach numbers

AMACH2(J,index) Jth Mach number.

ALPHA2 - Real (65,MAXAF2) Angles of attack, degrees

ALPHA2(I,index) Ith angle of attack

CLC2 - Real (65,9,MAXAF2) Lift coefficients

CLC2 (I, J, index) Ith angle of attack and Jth Mach number.

CD2 - Real (65,9,MAXAF2) Drag coefficient

CD2(I,J,index) Ith angle of attack and Jth Mach number.

CM2 - Real (65,9,MAXAF2) Moment coefficient

CM2(I,J,index) Ith angle of attack and Jth Mach number.

NAAF2 - Integer (2,MAXAF2) 8-character ds name of airfoil table in corresponding arrays.

NOTE: The angles of attack for each Mach number are from 0 to 360. The chordwise reference is the 1/4 chord.

COMMON Statement

```
      INTEGER AFDM, HEAD2, HEAD3
      COMMON /AFTAB/ AFDM (2), NAF2, NAF3,
     1        MAXAF2, MAXAF3, HEAD2 (20,5), CL2 (65,9,5),
     2        CD2 (65,9,5), CM2 (65,9,5), AMACH2 (9,5), ALPHA2
              (65,5),
     3        NMACH2 (5), NALFA2 (5), HEAD3 (20,1), CL3 (49,5,5,1),
     4        CD3 (49,5,5,1), CM3 (49,5,5,1), AMACH3 (5,1),
     5        ALPHA3 (49,1), DELTA3 (5,1) NMACH3 (1), NALFA3 (1),
     6        NODEL3 (1), NAAF2(2,5), NAAF3(2,1)
```

DATA Statement Initialization

```
      DATA MAXAF2, MAXAF3, NAF2, NAF3 /5,1,0,0/
      DATA AFDM /'AIRF','OIL'/
```

4.6.1.2 /XBG/ -

Purpose - To provide centralized communication for Base Variables, Global Variables, Global Constants, and Model Variables utilization. Those variables of interest to new Technology Module development are currently installed Global Constants, Global Variables, and Model Variables. These are given below:

Installed Global Constants

NAME	TYPE	VALUE	NAME	TYPE	VALUE
I-1	(I)	-1	NO	(Y)	'NO'
I0	(I)	0	YES	(Y)	'YES'
I1	(I)	1			
I2	(I)	2	R0	(R)	0.0
I3	(I)	3	R01	(R)	0.01
I4	(I)	4	R1	(R)	1.0
I5	(I)	5	R-1	(R)	-1.0
I6	(I)	6	R2	(R)	2.0
I7	(I)	7	R4	(R)	4.0
I8	(I)	8	R-45	(R)	-45.0
I9	(I)	9	R45	(R)	45.0
I10	(I)	10	R-90	(R)	-90.0
I15	(I)	15	R90	(R)	90.0
I20	(I)	20			
I25	(I)	25			
I30	(I)	30			
I35	(I)	35			
I40	(I)	40			
I45	(I)	45			
I50	(I)	50			
I60	(I)	60			

NAME	TYPE	VALUE
NCDFX	(I)	20 - Maximum number of DOFs
MCDFX	(I)	20 - Maximum number of DOFs except base
NCIDFX	(I)	20 - Maximum number of constraint equations
NXMAX	(I)	40 - Maximum number of aero stations
MAXEXT	(I)	6 - Maximum number of ds/AIRFOIL or sequential files combined in a model
NBX	(I)	9 - Maximum number of blades
NMODES	(I)	6 - Maximum number of elastic modes
NROTX	(I)	4 - Maximum number of rotors
NIX	(I)	12 - Maximum number of other implicit DOFs
NSX	(I)	20 - Maximum number of fuselage stations

Installed Global Variables

NAME	(TYPE, PROPERTY)	DESCRIPTION
RPM	(R, Single)	Rotor RPM (GE 0.0)
VSOUND	(R, Single)	Speed of sound (ft/sec) (GE 0.0)
RHO	(R, Single)	Air density ratio (GE 0.0)

Installed Model Variables

NAME	(TYPE, PROPERTY)	DESCRIPTION
NSDF	(I, Single)	Number of System DOFs

4.6.1.3 /XMODEL/ -

Purpose - To provide a centralized location for "model" variables; these include those used by Executive for controlling formation and utilization of a current ds/MODEL and also those technical variables related to the coupled system. Also included are maximum dimensioning variables for parameterization of code. Those listed below are saved/restored in ds/CASE formation and usage, and thus, can be used as input arguments for certain Solution Technical Modules as

described in the Solution Construction section. This is not an exhaustive list of /XMODEL/.

Model Variables

MAXC	- Maximum number of components in a model.
MTIT	- (18) 71-character description of model.
NAMEC	- (MAXC) names of components in model.
NAMEF	- (MAXC) names of forces in model.
NCOMP	- Number of components in current model (.GT. 0).
NOC	- (MAXC) occurrence number of component in model. 1 for first time this component referenced in model, etc.
NOF	- (MAXC) occurrence number of force in model.
NUAN	- (MAXC) user assigned number for component.

Coupled System Variables

NSCOX	- Maximum number of system implicit coefficients.
NSCO	- Number of system implicit coefficients.
SCO	- Real (NSCOX) system implicit coefficients.
SCODF	- Integer (NSCOX) system DOF corresponding to SCO.
NSDFX	- Maximum number of system DOFs.
NSDF	- Number of system DOFs.
SDFL	- Integer (NSDFX) literal part of system DOF names.
SDFI	- Integer (NSDFX) integer part of system DOF names.
NCDFX	- Maximum number of component DOFs.
NCDF	- (MAXC) number of component DOFs.
CDFLA	- Integer (NCDFX, MAXC) 4-character literal part of component DOF names.
CDFIA	- Integer (NCDFX,MAXC) integer part of component DOF names.
SM	- Real (NSDFX,NSDFX) system constant mass matrix.
SC	- Real (NSDFX,NSDFX) system constant damping matrix.
SK	- Real (NSDFX,NSDFX) system constant stiffness matrix.
SF	- Real (NSDFX) system constant force vector.
TRAN	- Integer (NCDFX, MAXC) transformation table.

4.6.1.4 <u>/XTM/</u> -

<u>Purpose</u> - Contains descriptive information about all installed Components, Forces, Solutions, and shared common blocks for use by the DYSCO Executive.

<u>Variables</u>

COMPONENT DESCRIPTION CONSTANTS

IARB - Type code for arbitrary component.

IROT - Type code for rotor component.

ICSYS - Type code for control system component.

ISTRUC - Type code for structural component.

IDROT - Type code for damaged rotor component.

COMPONENT INSTALLATION VARIABLES AND ARRAYS

MAXCOM - Maximum number of installed components without redimensioning.

NOCOM - Number of installed Components (increase by one). If exceeds MAXCOM, then dimensions must be increased.

CNAME - Integer (MAXCOM). 4-character Component names. Position of Component in this array is the "ordinal" of the Component. It is used, among other things, to access information in all installation arrays.

CTYPE - Integer (MAXCOM). Component type code.
- = 0 arbitrary
- = 1 rotor
- = 2 control system
- = 3 structural
- = 4 damaged rotor

CMAXO - Integer (MAXCOM). Maximum number of times this Component can occur within a given model.
- rotor - use 4 because of /ROT/ dimensioning
- other - use 20 because limited by /XMODEL/ dimensioning or use less than 20 if other restriction.

CNVF - Integer (MAXCOM). Number of forces valid for this Component.

CVFN - Integer (MAXCOM,MAXCOM). Names of forces valid for Component.

((CVFN (I,J), I=1,CNVF(J)), J=1,NOCOM)

CDES - Integer (6,MAXCOM). 24-character description of Component.

CNSCB - Integer (MAXCOM). Number of shared commons used by Component. (.LE. MAXCB1)

CNASCB - Integer (2,MAXCB1,MAXCOM). 6-character name of shared common blocks required for Component.

(((CNASCB (I,J,K) I=1,2), J=1,CNSCB(K)), K=1,NOCOM)

FORCE INSTALLATION VARIABLES AND ARRAYS

MAXFOM - Maximum number of installed Forces before arrays must be redimensioned.

NOFOM - Number of installed Forces (increase by one). If exceeds MAXFOM, then array dimensions must be increased.

FNAME - Integer (MAXFOM). 4-character Force names.

FMAXO - Integer (MAXFOM). Maximum number of times a Force can occur in a model. This is determined by how the code is written for a Force.

FDES - Integer (6,MAXFOM). 24-character description of Force

FNSCB - Integer (MAXFOM). Number of shared common blocks required by Force.

FNASCB - Integer (2,MAXCB1,MAXFOM). 6-character names of shared common blocks required by Force. (blanks if not apply)

(((FNASCB (I,J,K), I=1,2), J=1,FNSCB(K)),K=1,NOFOM)

SOLUTION INSTALLATION VARIABLES AND ARRAYS

MAXSOM - Maximum number of installed Solutions without redimensioning arrays.

NOSOM - Number of installed Solutions (increase by one). If exceeds MAXSOM, then array dimensions must be increased.

SNAME - Integer (MAXSOM). 4-character name of Solutions.

SDES - Integer (6,MAXSOM). 24-character solution descriptions.

SNSCB - Integer (MAXSOM). Number of shared common blocks required by Solution.

SNASCB - Integer (2,MAXCB1,MAXSOM). 6-character names of shared common blocks required by Solution.

(((SNASCB (I,J,K), I=1,2), J=1,SNSCB(K)), K=1,NOSOM)

SHARED COMMON BLOCKS INSTALLATION VARIABLES AND ARRAYS

MAXSCB - Maximum number of shared common blocks which may be installed without changing dimensions.

MAXCB1 - Maximum number of shared common blocks which any particular component/force/solution may require.

NUMSCB - Number of currently installed shared common blocks.

NAMSCB - Integer (2,MAXSCB). 6-character names of installed shared commons.

4.6.1.5 /TCI/ -

Purpose - Provides constants for oding convenience.

Variables

CFH - Conversion factor RAD/SEC to CYCLE/SEC (6.28319)

CFSI - Conversion factor SLUGS to LB-SEC**2/IN (12.0)

CFDR - Conversion factor DEGREES to RADIANS (.017453)

CFRRS - Conversion factor RPM to RAD/SEC (9.5493)

GA	- Gravitational acceleration (IN/SEC SQ) (386.4)	
LE	- Integer.	'LE'
LT	- Integer.	'LT'
GE	- Integer.	'GE'
GT	- Integer.	'GT'
EQ	- Integer.	'EQ'
NE	- Integer.	'NE'
YES	- Integer.	'YES'
NO	- Integer.	'NO'
AND	- Integer.	'AND'
OR	- Integer.	'OR'

4.6.1.6 <u>/XWORK/</u> -

<u>Purpose</u> - To provide reusable working storage to solution modules, reducing program storage requirements. The user can equivalence local vectors and matrices to addresses in a single large work storage array.

<u>Variables</u>

WS - (64420). Work storage array.

4.6.2 <u>Shared Common Blocks</u>

4.6.2.1 <u>Installations</u> -

<u>Common Block /XTM/</u> - A new shared common block must be described to the DYSCO Executive via master variables/arrays in /XTM/. Refer to paragraph 4.6.1.4 for descriptions. Those which must be changed are as follows:

NUMSCB, NAMSCB

<u>Block Module X-----</u> - A subroutine must be written to read/write the shared common on a file as part of a ds/CASE. The name should be

composed of X and the first 5 characters of the shared common name (uniqueness of name must also be ensured). The specifications are identical to those for the Component Block Module. See paragraph 4.3.2.5 for a description.

FORTRAN Call Statement - A call to the Block Module must be inserted in XCWRLC. Placement of the call is dependent on position of the name of the shared common in array NAMSCB. This placement is the ordinal of the shared common.

4.6.2.2 Installed Shared Common Blocks -

4.6.2.2.1 /ROT/.

Purpose - To provide access to rotor system variables by rotor, control system, and rotor aerodynamics modules:

CCEØA	CCEØC	CCEØD	
CCE1A	CCE1C	CCE1D	
CRE3A	CRE3C	CRE3D	CRE3L
CRD3A	CRD3C	CRD3D	CRD3L
CRR2A	CRR2C	CRR2D	
FRAØA	FRAØC		
FRA2A	FRA2C		
FRA3A	FRA3C		

XROT

4.6.2.2.2 /CONROT/.

Purpose - To provide access to control system variables by rotor, control system, and solution modules:

CCEØA
CCE1A
CRE3A

CRD3A

CRR2A

SSF3A

STH3A

STR3A

XCONRO

4.6.2.2.3 /TRIM/.

Purpose - To provide access to trim solution variables by fuselage, rotor, rotor aerodynamics, and trim solution modules:

CFM2A	CFM2D
CFM3A	CFM3D
CRE3A	
CRD3A	
CRR2A	
FRAØA	
FRA2A	
FRA3A	
STR3A	
STH3A	
TRIM	
XTRIM	

4.6.2.2.4 /ERD/.

Purpose - To provide access to rotor system variables, rotor blade parameters in particular, by elastic rotor and damaged elastic rotor modules:

CRE3A	CRE3C	CRE3D	CRE3L
CRD3A	CRD3C	CRD3D	CRD3L

XERD

4.6.2.2.5 /ELS/.

Purpose - To provide access to lifting surface variables by lifting surface structural and aerodynamic modules:

CLS2A	CLS2C	CLS2D
FLA2A	FLA2C	
XELS		

4.6.2.2.6 /GLOCOR/

Purpose - To provide access to parameters associated with the transformation of accelerations from the global reference coordinate system to component coordinate systems:

CCE1A	CRD3A	STH3A
CFM2A	CRE3A	STH4A
CFM3A	CRR2A	STR3A
CLG2A	CSF1A	
CLS2A	CSF3A	XGLOCO

4.7 UTILITIES

4.7.1 Base/Global Retrieval Utilities (XBG---).

4.7.1.1 Overview -

Purpose - The purpose of the XBG utilities is to provide solution, coefficient, and definition modules (S---A, C---C, C---D, F---C) the means to retrieve user input data and global constants. The Base Variable input data has been described in the corresponding input modules (S---I, C---I, F---I), with values being supplied by the user under control of the Executive input processor. It has been stored on a data set (ds/C--- or ds/F---). The Global

Variables required for input have been selected in the corresponding input modules from the list of Global Variables defined in COMMON /XBG/ and have been stored as part of a model. The Model Variables and Global Constants are static and are defined in COMMON/XBG/.

Any of these may be retrieved by calling the appropriate XBG utility. When an XBG retrieval utility is called, the name of the variable is supplied as an argument. The utility will search for this variable in the following order until found: (1) Base Variables for this solution, component, or force; (2) Global Variables for this model; (3) Global Constants; and (4) Model Variables.

Error Handling - There are no error codes returned from an XBG utility. This is based on the philosophy that any error code should always be checked by the caller routine upon return from the utility and, in the case of XBG utility usage, the increase in software complexity is not justified. Any error which does occur is classified as a "bug" which the technical module developer can correct. An example is specifying a variable name which does not have an entry in the Base Variable, Global Variable, Model Variable, or Global Constants table which clearly is a coding error. Upon encountering any error, the XBG utility will print an error message and abort the DYSCO execution.

The types of errors are as follows:

1. The variable name specified has no entry in the Base/Global/Model Variable or Global Constant tables
2. The incorrect utility is used to retrieve a particular type of value(s)
3. The value of the variable does not exist (i.e., the existence criteria were not satisfied, thus no value was supplied by the user)
4. Insufficient space has been allocated by the caller to receive a vector or matrix.

Naming Convention - The utility names are of the form XBGtp, where t stands for the type code of variable whose value is being retrieved, and p stands for whether it is a single item (S), a vector of items (V), or a matrix of items (M). The type codes (see paragraph 4.2.2.1) are: R, I, A, Y, D, X, S, MD, SD, and CD. Thus, to retrieve a variable which is described in the Base Variable table as a single integer, the utility XBGIS is used; to retrieve a matrix of real values, the XBGRM is used; and so on. The appropriate utility is determined by the description of the variable in the Base/Global/Model Variable or Global Constant table and not by any actual dimensions. For example, a variable described as a real vector may have an actual dimension of 1, but is still retrieved by XBGRV, and not by XBGRS.

Input Arguments - Commonly used input arguments are described below:

NAM1, NAM2 - Integers. Each containing 4 characters of the variable name.

MAX1, MAX2 - Integers. Maximum dimensions used for variable values if a vector or a matrix. For a vector, only MAX1 applies. MAX1 for a vector must be at least as great as the actual size of the vector. However, for a matrix, both apply and the values of MAX1 and MAX2 MUST BE the values used for dimensioning the variable in the coefficient or definition calling module. These are used by the utility to correctly store the values of a matrix whose actual dimensions are smaller than those used for dimensioning. Of course, these will usually be parameter variables (such as NSDFX) and not actual numbers.

Output Arguments - Commonly used output arguments are described below:

IA1,IA2 - Integers. Actual dimensions of vector or matrix. For a vector, only IA1 is used.

RVAL - Real. Used in real value retrieval.

Single - RVAL has a single value.

Vector - RVAL (1...IA1) values returned.

Matrix - RVAL (MAX1,MAX2) is assumed to be the dimensioning used.

RVAL (1...IA1, 1...IA2) values returned.

IVAL - Integer. Used in integer retrieval.

Single, vector, matrix dimensioning is similar to RVAL.

IALPHA - Integer. Used in alpha retrieval with 4 characters/word.

Alpha characters are left-justified and blank-filled.

Single, vector, matrix dimensioning is similar to RVAL.

IYN - Integer. Alpha "YES" or "NO" value(s); left-justified, blank-filled.

4.7.1.2 XBG Utilities - FORTRAN functions are noted. The other utilities are subroutines:

LOGICAL FUNCTION XBGE (NAM1, NAM2, EXIST)

Purpose: Test existence of variable

Input: NAM1, NAM2

Output: EXIST - Logical. True if variable exists, false if not. Function value same as EXIST.

XBGRS (NAM1, NAM2, RVAL)

Purpose: Retrieve single real value.

Input: NAM1, NAM2

Output: RVAL - Single real value.

XBGRV (NAM1, NAM2, MAX1 IA1, RVAL)

Purpose: Retrieve real vector.

Input: NAM1, NAM2, MAX1

Output: IA1

RVAL (1...IA1) - Vector of real values.

XBGRM (NAM1, NAM2, MAX1 MAX2, IA1, IA2, MCODE, RVAL)

Purpose: Retrieve real matrix.

Input: NAM1, NAM2, MAX1, MAX2

Output: IA1, IA2

RVAL (1...IA1, 1...IA2) - Real matrix values.

MCODE - Code for matrix type.

0 - Null (zero-filled) matrix

1 - Diagonal matrix

2 - Symmetric matrix

3 - General matrix (none of the above)

XBGIS (NAM1, NAM2, IVAL)

Purpose: Retrieve single integer value.

Input: NAM1, NAM2

Output: IVAL - Integer value.

XBGIV (NAM1, NAM2, MAX1 IA1, IVAL)

Purpose: Retrieve integer vector.

Input: NAM1, NAM2, MAX1

Output: IA1

IVAL (1...IA1) - Vector of integer values.

XBGIM (NAM1, NAM2, MAX1 MAX2, IA1, IA2, MCODE, IVAL)

Purpose: Retrieve integer matrix.

Input: NAM1, NAM2, MAX1, MAX2

Output: IA1, IA2

IVAL (1...IA1, 1...IA2) - Matrix of integer values.

MCODE - Code for matrix type.

0 - Null (zero-filled) matrix

1 - Diagonal matrix

2 - Symmetric matrix

3 - General matrix (none of the above)

XBGAS (NAM1, NAM2, IALPHA)

Purpose: Retrieve single alpha.

Input: NAM1, NAM2

Output: IALPHA - 4 character alpha value.

XBGAV (NAM1, NAM2, MAX1 IA1, IALPHA)

Purpose: Retrieve vector of alpha values.

Input: NAM1, NAM2, MAX1

Output: IA1

IALPHA (1...IA1) - Vector of alpha values (4 characters/word)

XBGDS (NAM1, NAM2, IALPHA IVAL)

Purpose: Retrieve single DOF name.

Input: NAM1, NAM2

Output: IALPHA - 4 character alpha part of DOF name

IVAL - Integer part of DOF name.

XBGDV (NAM1, NAM2, MAX1 IA1, IALPHA, IVAL)

Purpose: Retrieve vector of DOF names.

Input: NAM1, NAM2, MAX1

Output: IA1

IALPHA (1...IA1) - Vector of 4 character alpha part of DOF names.

IVAL (1...IA1) - Vector of integer part of DOF names.

XBGCDS (NAM1, NAM2, ICAT, IND)

Purpose: Retrieve single CD type value.

Input: NAM1, NAM2

Output: ICAT - Number of DOFs selected from single component (1).

IND - Index of component DOF from CDFLA, CDFIA array.

XBGCDV (NAM1, NAM2, MAXCAT, MAXIND, LCAT, ICAT, LIND, IND)

Purpose: Retrieve vector of CD type values

Input: NAM1, NAM2

MAXCAT - MAXC (maximum number of components).

MAXIND - Maximum number of indices in IND vector (maximum dimension).

Output LCAT - Number of entries in ICAT vector (number of components).

ICAT (1 ... LCAT) - Vector of number of DOFs selected from each component.

LIND - number of indices in IND vector.

IND - Vector of component DOF indices from CDFLA, CDFIA arrays.

XBGSDS (NAM1, NAM2, IND)

Purpose: Retrieve single SD type value.

Input: NAM1, NAM2

Output: IND - Index of system DOF from SDFL, SDFI array.

XBGSDV (NAM1, NAM2, MAXIND, LIND, IND)

Purpose: Retrieve vector of SD type values.

Input: NAM1, NAM2

MAXIND - Maximum number of indices in IND vector (maximum dimension).

Output: LIND - number of indices in IND vector.

IND - Vector of system DOF indices from SDFL, SDFI array.

XBGMDV (NAM1, NAM2, MAXCAT, MAXIND, LCAT, ICAT, LIND, IND)

Purpose: Retrieve vector of MD type values.

Input: NAM1, NAM2

MAXCAT - MAXC+1 (maximum number of components +1)

MAXIND - Maximum number of indices in IND vector (maximum dimension).

Output: LCAT - Number of entries in ICAT vector (number of components +1)

ICAT (1 ... LCAT) - Vector of number of DOFs selected from system followed by number of DOFs selected from each component.

LIND - number of indices in IND vector.

IND - Vector of system DOF indices from SDFL, SDFI array followed by component DOF indices from CDFLA, CDFIA arrays.

XBGYS (NAM1, NAM2, IYN)

Purpose: Retrieve YES/NO value.

Input: NAM1, NAM2

Output: IYN - Alpha "YES" or "NO" value left-justified

XBGYV (NAM1, NAM2, MAX1 IA1, IYN)

Purpose: Retrieve vector of YES/NO values.

Input: NAM1, NAM2, MAX1

Output: IA1

IYN (1...IA1) - Vector of "YES" or "NO" values left-justified.

XBGYM (NAM1 NAM2, MAX1, MAX2, IA1, IA2, MCODE, IVAL)

Purpose: Retrieve matrix of YES/NO values.

Input: NAM1, NAM2, MAX1, MAX2

Output: IA1, IA2

IVAL (1 ... IA1, 1 ... IA2) - Matrix of YES/NO values.

MCODE - Code for matrix type. Always 3, general.

XBGXS (NAM1, NAM2, INDEX ITYPE)

Purpose: Retrieve usage information about airfoil table

Input: NAM1, NAM2

Output: INDEX - Index to be used to retrieve airfoil values stored in /AFTAB/ arrays.

If ITYPE = 2, then INDEX = 1, 2, 3, 4, or 5

NOTE: For usage, see airfoil table common block /AFTAB/ in paragraph 4.6.1.1.

XBGSS (NAM1, NAM2, LUNSF)

Purpose: Retrieve usage information about sequential file.

Input: NAM1, NAM2

Output: LUNSF - Integer unit number of sequential file to be used in fortran READ/WRITE statements.

4.7.2 General Purpose Utilities

4.7.2.1 Overview -

Purpose - General purpose utilities are provided for convenience and efficiency and can be used by any technical module.

Error Handling - These utilities are, in general, written to validate input arguments at run-time. If an error is detected, the Executive will print an error message and terminate DYSCO.

Conventions - Arguments which are commonly used in the utilities are defined below:

IARRAY	- Integer array of values.
ARRAY	- Real array of values.
IVALUE	- Integer value.
VALUE	- Real value.
NUM	- Integer number of values in an array. (.GT. 0 unless stated otherwise)
NVAR1, NVAR2	- Integer. 8-character name of a variable used in error messages (4 characters/word).
T1F	- Integer. 2-character code for testing input value against test value1. Valid codes are: GE, GT, LE, LT, EQ.
T1V	- Integer or real, depending on utility. The test value against which input value will be tested in condition.

T2 - Logical. If .TRUE. then condition2 will be tested. If .FALSE. then condition 2 will not be tested and T2F, T2V will be ignored.

T2F - Integer. 2-character code for testing input value against test value2. Codes are: GE, GT, LE, LT, EQ.

T2V - Integer or real, depending on utility. The test value for condition2.

Forms for condition1 and condition2 are:

(input-value .T1F. T1V) .AND./.OR.
(input-value .T2F. T2V)

Example: (input-value .GE. 3) .AND.
(input-value .LT. 9)

OUT - Integer unit of output to terminal.

INP - Integer unit of input from terminal.

ERR - Logical. Return flag indicating result. If pre-initialization is required it is denoted as an input argument as well as output argument. This allows calling module to execute a sequence of tests and checking for error condition only at end.

4.7.2.2 General Utilities -

LOGICAL FUNCTION BLANKS (IARRAY, NUM)

Purpose: Test if array is all blank.

Input: IARRAY, NUM

Output: Function value .TRUE. if all are blank.

LOGICAL FUNCTION EQUALI (I1, I2, NUM)

Purpose: Test if two integer arrays are equal.

Input: I1, I2 - Integer arrays.

NUM - Number of elements. If 0 then considered equal.

Output: Function value .TRUE. if elements equal.

ISET (NUM, IARRAY, IVALUE)

Purpose: Set an array to single integer value.

Input: NUM - Number of array elements to be set.

IVALUE - Integer value.

Output: IARRAY

ITESTA (NVAR1, NVAR2, IVALUE, T1F, T1V, T2, T2F, T2V, OUT, ERR)

Purpose: Test an integer value against condition1 and (optional) condition2. Write error message if test not satisfied.

Input: NVAR1, NVAR2, IVALUE, T1F, T1V, T2, T2F, T2V, OUT, ERR

Output: ERR - .TRUE. if test fails. Unchanged if test succeeds.

ITESTO (NVAR1, NVAR2, IVALUE, T1F, T1V, T2F, T2V, OUT, ERR)

Purpose: Test an integer value against condition1 or condition2. If only one condition is used then second condition must be equal to first. If invalid, print error message.

Input: NVAR1, NVAR2, IVALUE, T1F, T1V, T2F, T2V, OUT, ERR

Output: ERR - .TRUE. if test fails. Unchanged if test succeeds.

XRMATW (UNIT, ITYP, ARRAY, MAX1, MAX2, IDIM1, IDIM2, NREC, NEXT)

Purpose: Print a real square matrix on either the OUT, RDF, or a UDF file.

Input: UNIT - Integer unit of output file.

ITYP - Integer matrix code.

0 - Null

1 - Diagonal

2 - Symmetric

3 - General

ARRAY - Real ARRAY(MAX1, MAX2). Matrix to be printed.

MAX1, MAX2 - Maximum dimensions of matrix.

IDIM1, IDIM2 - Actual dimensions of matrix to be printed.

NREC - Integer. Not used for writing to file OUT. Record to begin write to RDF or UDF.

Output: NEXT - Integer. Not used for writing to file OUT. Record following last record written to RDF or UDF.

MATINV (A, B, ISING, NACT, NMAX, IWORK1, IWORK2, RWORK1)

Purpose: Invert a real square matrix.

Input: A - Real matrix to be inverted.
NMAX - Maximum dimension of matrix.
NACT - Actual dimension of matrix.
IWORK1, IWORK2 - Integer arrays (NACT) for work space.
RWORK1 - Real array (NACT) for work space.

Output: B - Inverted matrix.
ISING - Integer result code.
0 - Inversion successful
9 - Singular matrix

RTESTA (NVAR1, NVAR2, VALUE, T1F, T1V, T2, T2F, T2V, OUT, ERR)

Purpose: Test a real value against condition1 and (optional) condition2. If invalid, print error message.

Input: NVAR1, NVAR2, VALUE, T1F, T1V, T2, T2F, T2V, OUT, ERR

Output: ERR - .TRUE. if test fails. Unchanged if test succeeds.

RTESTI (NVAR1, NVAR2, NUM, ARRAY, T1F, OUT, ERR)

Purpose: Test if a real array is (strictly or not strictly) increasing.

Input: NVAR1, NVAR2, NUM, ARRAY, T1F, OUT, ERR
T1F - Integer.
'GT ' - strictly increasing array
'GE ' - not strictly increasing array
(ARRAY(i) .GE. ARRAY(i-1))

Output: ERR - .TRUE. if test fails. Unchanged if test succeeds.

RTESTO (NVAR1, NVAR2, VALUE, T1F, T1V, T2F, T2V, OUT, ERR)

Purpose: Test a real value against condition1 or condition2. If invalid, print error message.

Input: NVAR1, NVAR2, VALUE, T1F, T1V, T2F, T2V, OUT, ERR

Output: ERR - .TRUE. if test fails. Unchanged if test succeeds.

LOGICAL FUNCTION VNO (IVALUE)

Purpose: Test if value is 'NO '.

Input: IVALUE

Output: Function value .TRUE. if test succeeds.

LOGICAL FUNCTION VYES (IVALUE)

Purpose: Test if value is 'YES'.

Input: IVALUE

Output: Function value .TRUE. if test succeeds.

YNC10 (NUM, IARRAY)

Purpose: Convert array of 'YES'/'NO' values to 1/0, respectively.

Input: NUM, IARRAY

Output: IARRAY - Integer array of 1/0 values.

LOGICAL FUNCTION YORN (NUM, IARRAY)

Purpose: Test if all array values are either 'YES', 'Y', 'NO', or 'N'.

Input: NUM, IARRAY

Output: Function value .TRUE. if test satisfied.

INTEGER FUNCTION XCONIA (I)

Purpose: Convert integer (0 through 99) to alpha form.

Input: I - Integer. (.GE. 0) and (.LE. 99)

Output: Function value is left-justified blank-filled alpha form.

XCOPYI (IFROM, ITO, NUM)

Purpose: Copy one integer array to another.

Input: IFROM - Integer. Array to be copied from.

NUM - Integer. Number of elements to copy. (.GE. 0)

Output: ITO - Integer. Copy of IFROM array.
Undetermined if NUM = 0.

XCOPYR (RFROM, RTO, NUM)

Purpose: Copy one real array to another.

Input: RFROM - Integer. Array to be copied from.

NUM - Integer. Number of elements to copy. (.GE. 0)

Output: RTO - Real. Copy of RFROM array.

XCOUNT (IARRAY, NUM, IVALUE, ICOUNT)

Purpose: Count the number of times an integer value is found in an array.

Input: IARRAY, IVALUE

NUM - Number of elements in IARRAY. (.GE.0)

Output: ICOUNT - Number of times value found

XFATAL (SUB1, SUB2, NVAR1, NVAR2, ITYP, NUM, IARRAY, ARRAY)

Purpose: Print fatal error message and terminate DYSCO. Typical use is due to erroneous variable value.

Input: SUB1, SUB2 - Integer. 8-character name of module calling XFATAL.

NVAR1, NVAR2 - Integer. 8-character name of variable in error.

ITYP - Integer. Type of errorful variable.

1 - Integer

2 - Real

3 - Alpha

NUM - Integer. Number of values to be printed.

IARRAY(NUM) or ARRAY(NUM) depending on ITYP

IARRAY - Integer. If ITYP is integer or alpha, then this array is assumed errorful and will be printed.

ARRAY - Real. If ITYP is real, then this array printed

Output: None

XFIND (IARRAY, NUM, IVALUE, IOC, IORD)

Purpose Find specified occurrence of integer value in an integer array.

Input: IARRAY, IVALUE

NUM - Integer. Number of values in array (.GE. 0)

IOC - Integer. Occurrence number to be found (.GT. 0)

Output: IORD - Integer. Index in array where IOCth occurrence found. If NUM = 0 then IORD = 0.

LOGICAL FUNCTION XVECR (ICODE, ARRAY, NUM, MSG, IPOS)

Purpose: Test if real array (vector) is increasing/decreasing and if not, return subscript of first violating element.

Input: ICODE - Integer test code.

1 - Increasing (ARRAY(i) .GE. ARRAY(i-1))

2 - Decreasing (ARRAY(i) .LE. ARRAY(i-1))

ARRAY, NUM

MSG - Logical. .TRUE. then print error message and element if test fails. .FALSE. then no print.

Output: IPOS - Integer. If test fails, then contains subscript to first element in ARRAY violating test. Function value .TRUE. if test succeeds.

LOGICAL FUNCTION XVECI (ICODE, ARRAY, NUM, LSG, IPOS)

Purpose: Test if integer array (vector) is increasing/decreasing and if not, return subscript of first violating element.

<u>Input</u>:	ICODE	- Integer test code. 1 - Increasing (IARRAY(i) .GE. IARRAY(i-1)) 2 - Decreasing (IARRAY(i) .LE. IARRAY(i-1))
	IARRAY, NUM	
	MSG	- Logical. .TRUE. then print error message if test fails. .FALSE. then no print.
<u>Output</u>:	IPOS	- Integer. If test fails, then contains subscript to first element in IARRAY violating test.

5.0 ILLUSTRATIVE APPLICATIONS

Applications which illustrate program capabilities will be discussed. The setup and modification of a basic helicopter simulation will be presented and may suggest extensions of the applications or enhancements to the program.

All DYSCO capabilities could be applied to the assembly and analysis of a few variations of a given helicopter model. The definition and coupling of the components of a helicopter ground resonance model will be discussed. This basic model will then be modified to simulate damage conditions, and additional components will be incorporated to simulate more complex configurations.

5.1 COUPLING OF STANDARDIZED DEGREES OF FREEDOM

CFM2, CRR2, CRE3, CCEØ, and CCE1 have standardized degree of freedom names. That is, the degree of freedom names are formulated automatically. Coupling one or more rotors with optional control systems to a rigid or elastic fuselage only requires that the user supply the component information as requested. No further consideration need be given to the couplings.

A simple example is the input for the rotor and fuselage components of a ground resonance model. Pertinent inputs are shown below:

```
CRR2:

-----
IBETA  (Y OR N)
     BLADE FLAPPING DOF
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                               *
*    BLADE MUST HAVE AT LEAST ONE OF FLAP, LAG OR PITCH DOF     *
*                                                               *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
ENTER 1 Y OR N VALUE
N

-----
IZETA      (Y OR N)
     BLADE LAG DOF
ENTER 1 Y OR N VALUE
Y
```

```
-----
ITHET       (Y OR N)
     BLADE PITCH DOF
ENTER 1 Y OR N VALUE
N

-----
IX          (Y OR N)
     HUB TRAN DOF - LONG
ENTER 1 Y OR N VALUE
N

-----
IY          (Y OR N)
     HUB TRAN DOF - LAT
ENTER 1 Y OR N VALUE
Y

-----
IZ          (Y OR N)
     HUB TRAN DOF - AXIAL
ENTER 1 Y OR N VALUE
N

-----
IAX         (Y OR N)
     HUB ANGL DOF - ROLL
ENTER 1 Y OR N VALUE
N

-----
IAY         (Y OR N)
     HUB ANGL DOF - PITCH
ENTER 1 Y OR N VALUE
N

-----
IAZ    (Y OR N)
     HUB ANGL DOF - YAW
ENTER 1 Y OR N VALUE
N

-----
NB     (INTEGER)
     NUMBER OF BLADES
ENTER  1 INTEGER VALUE(S)
2
```

<u>CFM2</u>:

```
-----
RBM          (Y OR N)
     RIGID BODY MODES
ENTER 1 Y OR N VALUE
Y
```

```
-----
IXCG        (Y OR N)
     LONGITUDINAL
ENTER 1 Y OR N VALUE
N

-----
IYCG       (Y OR N)
     LATERAL
ENTER 1 Y OR N VALUE
Y

-----
IZCG        (Y OR N)
     VERTICAL
ENTER 1  Y OR N VALUE
N

-----
IROLL       (Y OR N)
     ROLL
ENTER 1 Y OR N VALUE
Y

-----
IPTCH       (Y OR N)
     PITCH
ENTER 1 Y OR N VALUE
N

-----
IYAW        (Y OR N)
     YAW
ENTER 1 Y OR N VALUE
N

-----
NMODE       (INTEGER)
     NO. OF ELASTIC MODES
ENTER  1 INTEGER VALUE(S)
0

-----
NR          (INTEGER)
     NO. OF ROTORS
ENTER  1 INTEGER VALUE(S)
1
-----
NROT        (INTEGER)
     ROTOR NUMBERS
ENTER  1 INTEGER VALUE(S)
1
```

```
-----
XROT        (REAL)
     ROTOR STATIONS
ENTER  1 REAL VALUE(S)
?
-9.9

-----
ZROT        (REAL)
     ROTOR VERTICAL HT
ENTER  1 REAL VALUE(S)
?
84.24

-----
ASF    (REAL)
     FWD SHAFT ANGLE
ENTER  1 REAL VALUE(S)
?
6

-----
ASL    (REAL)
     LAT SHAFT ANGLE
ENTER  1 REAL VALUE(S)
?
4

-----
IX     (Y OR N)
     HUB TRAN DOF - LONG
ENTER  1 Y OR N VALUE
N

-----
IY    (Y OR N)
     HUB TRAN DOF - LAT
ENTER  1 Y OR N VALUE
Y

-----
IZ    (Y OR N)
     HUB TRAN DOF - AXIAL
ENTER 1 Y OR N VALUE
N

-----
IAX    (Y OR N)
     HUB ANGL DOF - ROLL
ENTER  1 Y OR N VALUE
N
```

```
-----
IAY     (Y OR N)
     HUB ANGL DOF - PITCH
ENTER  1 Y OR N VALUE
N
-----
IAZ   (Y OR N)
     HUB ANGL DOF - YAW
ENTER  1 Y OR N VALUE
N
-----
NI    (INTEGER)
     NO. OTHER IMPLCT DOF
ENTER  1 INTEGER VALUE(S)
0
```

The CRR2 input specifies the lag degree of freedom for two blades and the lateral translation degree of freedom for the hub. Lateral translation and roll have been specified for the fuselage. No elastic degrees of freedom have been specified for the fuselage. In addition, the rotor location and orientation, and the hub degrees of freedom to be coupled to the fuselage are specified in the CFM2 input. The hub degrees of freedom will automatically be expressed as linear combinations of the fuselage degrees of freedom when the model is assembled and run.

In the system equations for these two components, the degrees of freedom would be:

ZETA11ØØ
ZETA12ØØ
YCG1ØØØ
ROLL1ØØØ

5.2 CSF1 APPLICATION

The input for CSF1 is simply the constant coefficients of a general set of linear differential equations, making CSF1 useful for many applications. A

landing gear representation for the ground resonance model may easily be formulated using CSF1. Figure 27 shows a schematic of the complete ground resonance model: rotor, rigid fuselage, and landing gear. The landing gear consists of two main gear and a tail gear, each with three degrees of freedom constrained by springs and dampers representing tire and strut stiffness and strut damping. The parameters of the three landing gear systems can be input separately as follows:

```
NEW
NEW MODEL  (Y OR N)
N
COMPONENT, FORCE, OR N
CSF1
DATA SET
LMAIN
SAVE FILE
U1

COMPONENT CSF1. FINITE ELEMENT

BEGIN INPUT
DESCRIPTION (UP TO 71 CHARACTERS)
LEFT MAIN LANDING GEAR

-----
NCDF        (INTEGER)
     NUMBER OF DOF
ENTER  1 INTEGER VALUE(S)
3

-----
CDFLI       (DOF)
     DOF NAME
ENTER  3 DOF VALUES  (A4,I4) ONE PER LINE
X2
Y2
Z2

-----
CM    (REAL)
     MASS MATRIX VALUES
TYPE MATRIX
(0=NULL), (1=DIAGONAL), (2=SYMMETRIC), (3=GENERAL)
0
```

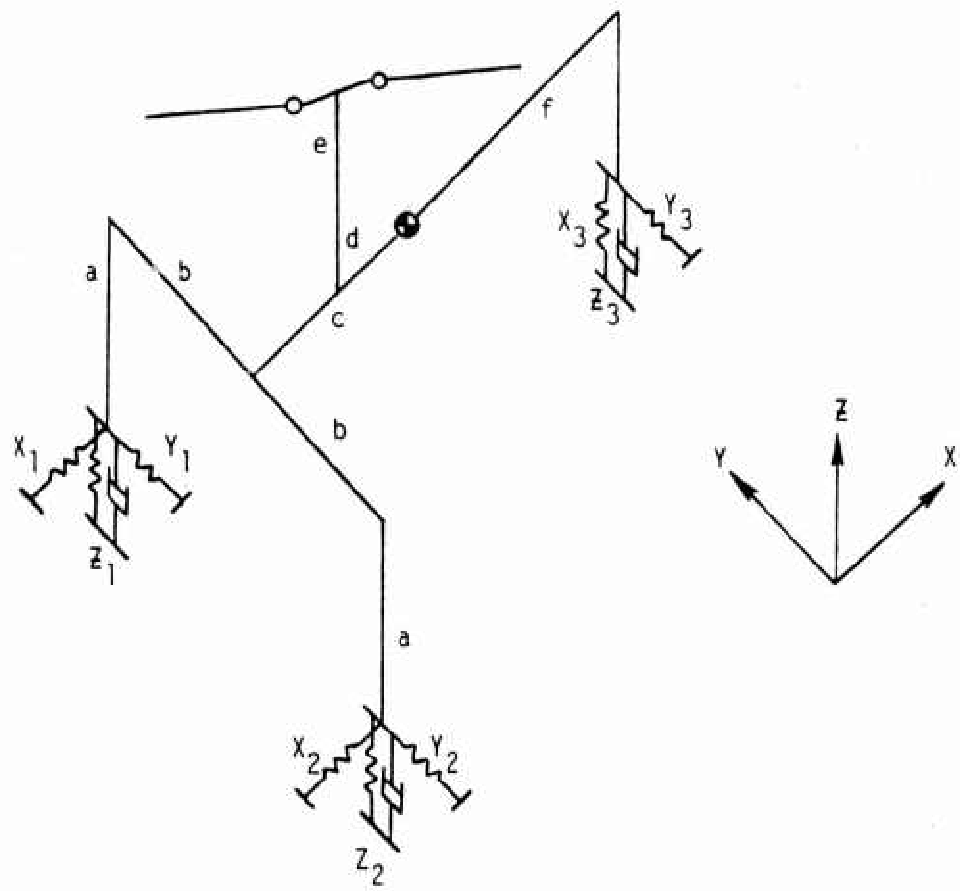


Figure 27. Ground Resonance Model.

```
-----
CC    (REAL)
     DAMPING MATRX VALUES
TYPE MATRIX
(0=NULL), (1=DIAGONAL), (2=SYMMETRIC), (3=GENERAL)
1
INPUT  3 DIAGONAL REAL VALUES
?
0 0 66.67

-----
CK     (REAL)
     STFFNESS MTRX VALUES
TYPE MATRIX
(0=NULL), (1=DIAGONAL), (2=SYMMETRIC), (3=GENERAL)
1
INPUT  3 DIAGONAL REAL VALUES
?
2500 2965 781.6

-----
CF     (REAL)
     FORCE VECTOR VALUES
NULL VECTOR  (Y OR N)
Y
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  *
INPUT FOR COMPONENT CSF1.FINITE ELEMENT

1  NCDF  - NUMBER OF DOF  =  3
2  CDFLI - (DOF) DOF NAME
           X2     0  Y2     0  Z2     0
3  CM    - (REAL) MASS MATRIX VALUES
            NULL MATRIX
4  CC    - (REAL) DAMPING MATRX VALUES
            DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
     0.00000E+00   0.00000E+00    6.66700E+01
5  CK    - (REAL) STFFNESS MTRX VALUES
            DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
     2.50000E+03   2.96500E+03    7.81600E+02
6  CF    - REAL FORCE VECTOR VALUES
     0.00000E+00   0.00000E+00     0.00000E+00
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  * *


RE-ENTER (Y OR N)
N
DATA SET LMAIN  FOR CSF1 SAVED ON U1
COMPONENT, FORCE, OR N
N
COMMAND
```

```
LIST
DATA SET
RMAIN
DATA MEMBER
CSF1
RMAIN  /CSF1  ON FILE U1

* * * * * * * * * * * * *  RMAIN  /CSF1  * * * * * * * * * * * * *

RIGHT MAIN LANDING GEAR

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INPUT FOR COMPONENT CSF1. FINITE ELEMENT

1  NCDF   - NUMBER OF DOF   =   3
2  CDFLI  - (DOF) DOF NAME
            X1      0  Y1      0  Z1      0
3  CM     - (REAL) MASS MATRIX VALUES
            NULL MATRIX
4  CC     - (REAL) DAMPING MATRX VALUES
            DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
            0.00000E+00    0.00000E+00   6.66700E+01
5  CK     - (REAL) STFFNESS MTRX VALUES
            DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
            2.50000E+03    2.73000E+03   7.70300E+02
6  CF     - FORCE VECTOR VALUES
            0.00000E+00    0.00000E+00   0.00000E+00
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

LIST COMPLETE
COMMAND

LIST
DATA SET
TAIL
DATA MEMBER
CSF1
TAIL  /CSF1  ON FILE U1

* * * * * * * * * * *  TAIL  /CSF1  * * * * * * * * * * * * * * *

TAIL LANDING GEAR

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INPUT FOR COMPONENT CSF1. FINITE ELEMENT

1  NCDF  -  NUMBER OF DOF       =       3
2  CDFLI -  (DOF)  DOF NAME
            X3      0  Y3      0  Z3      0
```

```
3  CM     -  (REAL) MASS MATRIX VALUES
             NULL MATRIX
4  CC     -  (REAL) DAMPING MATRX VALUES
             DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
             0.00000E+00    0.00000E+00    2.91700E+01
5  CK     -  (REAL) STFFNESS MTRX VALUES
             DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
             0.00000E+00    6.64700E+02    3.25800E+02
6  CF     -  FORCE VECTOR VALUES
             0.00000E+00    0.00000E+00    0.00000E+00
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

LIST COMPLETE
COMMAND
```

5.3 IMPLICIT COUPLING - CLC1

At this point in the ground resonance simulation, it remains for the landing gear to be coupled to the fuselage. This is accomplished by expressing the landing gear degrees of freedom in terms of the fuselage degrees of freedom and expressing those relationships in a CLC1 component.

For small angles, the displacement at any point on a rigid body is equal to the translation vector of the CG plus the vector cross product of the rigid body angular displacement and the radius vector from the CG to the given point.

$$\begin{Bmatrix} X_1 \\ Y_1 \\ Z_1 \end{Bmatrix} = \begin{Bmatrix} \mathrm{XCG} \\ \mathrm{YCG} \\ \mathrm{ZCG} \end{Bmatrix} + \begin{vmatrix} i & j & k \\ \mathrm{ROLL} & \mathrm{PTCH} & \mathrm{YAW} \\ -c & b & -a \end{vmatrix} = \begin{Bmatrix} \mathrm{XCG} - a \cdot \mathrm{PTCH} - b \cdot \mathrm{YAW} \\ \mathrm{YCG} + a \cdot \mathrm{ROLL} - c \cdot \mathrm{YAW} \\ \mathrm{ZCG} + b \cdot \mathrm{ROLL} + c \cdot \mathrm{PTCH} \end{Bmatrix}$$

$$\begin{Bmatrix} X_2 \\ Y_2 \\ Z_2 \end{Bmatrix} = \begin{Bmatrix} \mathrm{XCG} \\ \mathrm{YCG} \\ \mathrm{ZCG} \end{Bmatrix} + \begin{vmatrix} i & j & k \\ \mathrm{ROLL} & \mathrm{PTCH} & \mathrm{YAW} \\ -c & -b & -a \end{vmatrix} = \begin{Bmatrix} \mathrm{XCG} - a \cdot \mathrm{PTCH} + b \cdot \mathrm{YAW} \\ \mathrm{YCG} + a \cdot \mathrm{ROLL} - c \cdot \mathrm{YAW} \\ \mathrm{ZCG} - b \cdot \mathrm{ROLL} + c \cdot \mathrm{PTCH} \end{Bmatrix}$$

$$\begin{Bmatrix} X_3 \\ Y_3 \\ Z_3 \end{Bmatrix} = \begin{Bmatrix} \mathrm{XCG} \\ \mathrm{YCG} \\ \mathrm{ZCG} \end{Bmatrix} + \begin{vmatrix} i & j & k \\ \mathrm{ROLL} & \mathrm{PTCH} & \mathrm{YAW} \\ f & 0 & -a \end{vmatrix} = \begin{Bmatrix} \mathrm{XCG} - a \cdot \mathrm{PTCH} \\ \mathrm{YCG} + a \cdot \mathrm{ROLL} + f \cdot \mathrm{YAW} \\ \mathrm{ZCG} - f \cdot \mathrm{PTCH} \end{Bmatrix}$$

Considering only lateral displacements:

```
LIST
DATA SET
GCGLAT
DATA MEMBER
CLC1
GCGLAT  /CLC1       ON FILE U1

* * * * * * * * * * * *   GCGLAT  /CLC1  * * * * * * * * * * * * * *

COUPLE LANDING GEAR TO FUSELAGE LATERAL

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INPUT FOR COMPONENT CLC1. LINEAR CONSTRAINTS

1  NCDF    - NUMBER OF DOF            =       2
2  CDFLI   - DOF NAMES                =    YCG 1000   ROLL1000
3  NCIDF   - # OF CONSTRAINT EQNS     =       9
4  CIDFLI  - (DOF) IMPLICIT DOF NAME
           X2      0  Y2     0  Z2     0  X1     0  Y1     0
           Z1      0  X3     0  Y3     0  Z3     0
5  COEF    - (REAL) COEFFICIENT MATRIX

     GENERAL MATRIX

ROW  1          NULL ROW
ROW  2
   1.00000E+00       7.57600E+01
ROW  3
   0.00000E+00      -6.60000E+01
ROW  4          NULL ROW
ROW  5
   1.00000E+00       7.57600E+01
ROW  6
   0.00000E+00       6.60000E+01
ROW  7          NULL ROW
ROW  8
   1.00000E+00       7.57600E+01
ROW  9          NULL ROW
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

LIST COMPLETE
COMMAND
```

The complete ground resonance model is listed below, including degree of freedom details.

```
RUN
MODEL NAME (DATA SET)
GRLAT
LIST MODEL SUMMARY (Y OR N)
Y

* * * * * * * * * * * * *   MODEL GRLAT  * * * * * * * * * * * * * *

GROUND RESONANCE LATERAL

     INDEX    COMP        NO.        DATA SET       FORCE       DATA SET

       1      CRR2        1          ROT            NONE
       2      CFM2        1          FUS            NONE
       3      CSF1                   LMAIN          NONE
       4      CSF1                   RMAIN          NONE
       5      CSF1                   TAIL           NONE
       6      CLC1                   GCGLAT         NONE


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                              GLOBAL VARIABLES
NO INPUT REQUIRED
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

TEMPORARY RUN EDIT OF ANY COMPONENT/FORCE INPUT (Y OR N)
N

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
DETAILS (Y OR N)
Y
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                        COMPONENT DOF/SYSTEM DOF
                1, 6              2, 7              3, 8

1    CRR2       ZETA1100          ZETA1200          YHUB1000
                 (    1)           (    2)           (   -1)

2    CFM2       YCG 1000          ROLL1000
                 (    3)           (    4)

3    CSF1       X2       0        Y2       0        Z2       0
                 (    0)           (   -3)           (   -5)

4    CSF1       X1       0        Y1       0        Z1       0
                 (    0)           (   -6)           (   -8)
```

```
5    CSF1      X3     0      Y3     0      Z3     0
               (    0)       (    -9)      (     0)

6    CLC1      YCG 1000      ROLL1000
               (    3)       (     4)

          SYSTEM DOF
          1    ZETA1100
          2    ZETA1200
          3    YCG 1000
          4    ROLL1000
```

IMPLICIT COEFFICIENTS

I	COEF	DOF	I	COEF	DOF
1	9.976E-01	YCG 1000	6	1.000E+00	YCG 1000
2	-8.403E+01	*ROLL1000	7	7.576E+01	*ROLL1000
3	1.000E+00	YCG 1000	8	6.600E+01	*ROLL1000
4	7.576E+01	*ROLL1000	9	1.000E+00	YCG 1000
5	-6.600E+01	*ROLL1000	10	7.576E+01	*ROLL1000

```
PRINT MATRICES  (Y OR N)
N
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
SOLUTION OR N
N
COMMAND
```

In the details the positive integers in parentheses under the degree of freedom names indicate system degrees of freedom, the negative integers index the implicit coefficients, and zero indicates that a degree of freedom has been eliminated (implicit coefficients are zero). For example, -1 for YHUB1000 corresponds to the 1 in the I column of the implicit coefficients. YHUB1000 is replaced by

(9.976E-01 X YCG 1000) + (-8.403E+01 X ROLL1000)

An asterisk identifies the last coefficient of a combination. Note that the rotor shaft angles have been accounted for.

5.4 CES1 APPLICATION

In the ground resonance model, the blade lag degree of freedom is specified for the rotor. A spring and a damper can be specified for that degree of freedom in the CRR2 input. Alternatively, damped elastic blade lag stops may be simulated using CES1. The blade is allowed to swing freely through a given arc before impinging against a spring-damper.

CES1 input is shown below:

```
LIST
DATA SET
STOP
DATA MEMBER
CES1
STOP     /CES1         ON FILE U1
* * * * * * * * * * * * *    STOP      /CES1  * * * * * * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INPUT FOR COMPONENT CES1. ELASTIC STOP

     1    MCDF     - # of DOF-EXCEPT BASE=          2
     2    CDFLI    - DOF NAMES           =    ZETA1100  ZETA1200
     3    BASE     - EXISTNCE OF BASE DOF=       NO
     4    C1       - UPPER DAMPING COEFF =     6.28000E+02
     5    C2       - LOWER DAMPING COEFF =     6.28000E+02
     6    K1       - UPPER SPRING COEFF  =     1.06272E+04
     7    K2       - LOWER SPRING COEFF  =     1.06272E+04
     8    DELT1    - UPPER GAP SIZE      =     1.74500E-01
     9    DELT2    - LOWER GAP SIZE      =     1.74500E-01
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

LIST COMPLETE
COMMAND
```

Upper and lower correspond to positive and negative angular displacement, respectively. The blade is allowed to swing freely through ± .1745 rad (± 10 deg) before rebounding off a torsional spring and damper.

5.5 DAMAGE SIMULATION

The simulation of damage or modification to a system can be done four different ways:

1. Use the temporary editor to modify a component or components at run time.

2. Use the editor to replace a component with another reflecting the damage condition. For example, the CSF1 component representing the right main landing gear in the ground resonance model could be replaced by a component representing a failed strut.

```
LIST
DATA SET
RMAINFS
DATA MEMBER
CSF1
RMAINFS  /CSF1      ON FILE U1

* * * * * * * * * * * * *     RMAINFS  /CSF1   * * * * * * * * * * *

RIGHT MAIN LANDING GEAR, FAILED STRUT

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INPUT FOR COMPONENT CSF1. FINITE ELEMENT

     1    NCDF      - NUMBER OF DOF     =         3
     2    CDFLI     - (DOF) DOF NAME
                 X1     0  Y1     0  Z1     0
     3    CM         - (REAL) MASS MATRIX VALUES
                 NULL MATRIX
     4    CC         - (REAL) DAMPING MATRX VALUES
                 NULL MATRIX
     5    CK         - (REAL) STIFFNESS MTRX VALUES
                DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
              2.50000E+03    2.95000E+03    3.34400E+03
     6    CF         - FORCE VECTOR VALUES
              0.00000E+00    0.00000E+00    0.00000E+00
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

LIST COMPLETE
COMMAND
```

In this case the damage has been modeled by eliminating the damping in the strut and modifying the combined stiffness of the gear and tires.

3. The last two methods of damage simulation make use of the generality of the CSF1 component. A CSF1 component can be used to modify the constant coefficients of the system equations by giving it the desired system degrees of freedom and inputting values which will be added directly to the system coefficients when the component is incorporated in the model.

4. All components, regardless of input, are expressed internally by DYSCO as coefficient matrices which are combined in the system equations for a model. A CSF1 component can be given degrees of freedom of other components and be used as above to modify coefficients.

   For example, if the blade pitch degree of freedom is added to the CRR2 component of the ground resonance model, a control rod component can also be used with the model, each blade being coupled with a control rod. However, the control rod input is constrained to giving all of the rods identical physical characteristics. CSF1 can be used to simulate a single failed or partially failed control rod by negating the stiffness for a single control rod degree of freedom as shown below:

```
LIST
DATA SET
FAIL PL
DATA MEMBER
CSF1
FAIL PL  /CSF1      ON FILE U1
* * * * * * * * * * * * * *   FAIL PL  /CSF1  * * * * * * * * * * *

FAIL PITCH LINK BLADE 1

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
INPUT FOR COMPONENT CSF1. FINITE ELEMENT

     1    NCDF       - NUMBER OF DOF      =         1
     2    CDFLI      - DOF NAME           =    RODR1100
     3    CM         - (REAL) MASS MATRIX VALUES
               NULL MATRIX
     4    CC         - (REAL) DAMPING MATRX VALUES
               NULL MATRIX
     5    CK         - (REAL) STFFNESS MTRX VALUES
               DIAGONAL MATRIX (DIAGONAL VALUES PRINTED)
          -4.62800E+03
     6    CF         - FORCE VECTOR VALUES =  0.00000E+00
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

LIST COMPLETE
COMMAND
```

Note that RODR1100 does not appear in the system equations (see paragraph 3.1.2.3, Implicit Degrees of Freedom). The stiffness change and transformation is carried out automatically during model formulation.

6.0 ILLUSTRATION OF AH-1G ANALYSIS

In this section, the details of AH1G-35A/MODEL and its associated components and force modules are listed. Then, a trim sample run and a typical sample output will be presented. It is suggested that the reader refer to Volume I, Theoretical Manual, for detailed discussion.The material presented will be in the following order:

1. AH1G-35A/MODEL
2. B2Z1T2/CRE3
3. FCT1.65/FRA3
4. 3ØØØ/CCEØ
5. COUPLE/CLC1
6. 8300-4/CFM2
7. AH1G 16.5/FFC2
8. AH1G-35A Trim run, sample input and output.

```
LIST
DATA SET
AH1G-35A
DATA MEMBER
MODEL
AH1G-35A/MODEL     ON FILE U1

************************       MODEL AH1G-35A  ******************

AH1G TRIM

INDEX    COMP     NO.    DATA SET          FORCE   DATA SET

  1      CRE3      1     B2Z1T2            FRA3    FCT1.65
                                      REQUIRED DS/DM= AFD161   /AIRFOIL
  2      CCE0      1     3000              NONE
  3      CLC1            COUPLE            NONE
  4      CFM2      1     8300-4            FFC2    AH1G16.5

***************************************************************

***************************************************************
                              GLOBAL   VARIABLES

  1 VSOUND    - SOUND VELOCITY        =   1.13800E+03
  2 RHO       - AIR DENSITY RATIO     =   8.79000E-01
***************************************************************


LIST COMPLETE
COMMAND
```

```
LIST
DATA SET
B2Z1T2
DATA MEMBER
CRE3
B2Z1T2  /CRE3     ON FILE U1

***************       B2Z1T2  /CRE3      ***************

ELASTIC ROTOR WITH 2 OP, 1 IP, 2 TOR MODES

*******************************************************************************
INPUT FOR     ROTOR      COMPONENT CRE3. ROTOR ELASTIC BLADES

  1 JV        - INPLANE DOF          =         YES
  2 JW        - OUTPLANE DOF         =         YES
  3 JP        - TORSION DOF          =         YES
  4 JS        - SHAFT PERTURBED DOF  =         NO
  5 JX        - XHUB(LONG) DOF       =         YES
  6 JY        - YHUB(LAT) DOF        =         NO
  7 JZ        - ZHUB(AXIAL) DOF      =         YES

  8 JAX       - ALFX(ROLL) DOF       =         YES
  9 JAY       - ALFY(PTCH) DOF       =         YES
 10 JAZ       - ALFZ(YAW) DOF        =         NO
 11 NV        - NO. OF INPLANE MODES=             1
 12 NW        - NO. OF OUTPLNE MODES=             2
 13 NP        - NO. OF TORSION MODES=             2
 14 NB        - NO. OF BLADES        =            2
 15 PHL       - PTCH HORN LNGTH (IN)= -9.25000E+00
 16 PHSTA     - PTCH HORN STA (IN)   =  1.41000E+01
 17 NX        - NO. OF STATIONS      =           20
 18 ITYP      - MODE INPUT 1 OR 2    =            1
 19 X         - (REAL) STATIONS
              0.00000E+00  1.50000E+01  4.50000E+01  5.27900E+01
              5.28000E+01  6.29000E+01  8.27000E+01  1.01700E+02
              1.21500E+02  1.38600E+02  1.50500E+02  1.63700E+02
              1.78200E+02  1.91400E+02  2.05150E+02  2.18350E+02
              2.31250E+02  2.45500E+02  2.57400E+02  2.64000E+02
 20 NIP       - INPLANE HINGE STA    =            1
 21 NOP       - OUTPLANE HINGE STA   =            1
 22 NTOR      - PTCH BEARING STA     =            1
 23 VPP       - (REAL) 2ND DERIVATIVE OF IP
             GENERAL MATRIX
```

```
ROW    1
       3.64850E-05
ROW    2
       9.24470E-05
ROW    3
       1.53210E-05
ROW    4
       1.44120E-05
ROW    5
       1.44120E-05
ROW    6
       2.58380E-05
ROW    7
       2.22030E-05
ROW    8
       1.79180E-05
ROW    9
       1.61000E-05
ROW   10
       1.37630E-05
ROW   11

       1.23350E-05
ROW   12
       1.02570E-05
ROW   13
       8.69940E-06
ROW   14
       7.01150E-06
ROW   15
       4.67430E-06
ROW   16
       2.85650E-06
ROW   17
       1.55810E-06
ROW   18
       5.19360E-07
ROW   19      NULL ROW
ROW   20      NULL ROW
24 VP        - (REAL) 1ST DERIVATIVE OF IP
              GENERAL MATRIX

ROW    1      NULL ROW
ROW    2
```

9.67000E-04
ROW 3
2.58350E-03
ROW 4
2.69930E-03
ROW 5
2.69950E-03
ROW 6
2.90270E-03
ROW 7
3.37830E-03
ROW 8
3.75950E-03
ROW 9
4.09630E-03
ROW 10
4.35160E-03
ROW 11
4.50690E-03
ROW 12
4.65600E-03
ROW 13

4.79340E-03
ROW 14
4.99710E-03
ROW 15
4.97750E-03
ROW 16
5.02720E-03
ROW 17
5.05560E-03
ROW 18
5.07050E-03
ROW 19
5.07350E-03
ROW 20
5.37500E-03

25 V - (REAL) INPLANE MODE SHAPES
GENERAL MATRIX

ROW 1 NULL ROW
ROW 2
7.25[illegible]00E-03
ROW 3

```
          6.05100E-02
ROW        4
          8.10800E-02
ROW        5
          8.11100E-02
ROW        6
          1.09400E-01
ROW        7
          1.71590E-01
ROW        8
          2.39400E-01
ROW        9
          3.17100E-01
ROW       10
          3.89400E-01
ROW       11
          4.42100E-01
ROW       12
          5.02500E-01
ROW       13
          5.71100E-01
ROW       14

          6.35000E-01
ROW       15
          7.03000E-01
ROW       16
          7.68900E-01
ROW       17
          8.34000E-01
ROW       18
          9.06100E-01
ROW       19
          9.66500E-01
ROW       20
          1.00000E+00
26 WFF        - (REAL) 2ND DERIVATIVE OF OF
           GENERAL MATRIX

ROW        1
          0.00000E+00   3.91000E-05
ROW        2
          0.00000E+00   5.00700E-05
ROW        3
          0.00000E+00   1.46800E-05
```

```
ROW     4
        0.00000E+00   1.26500E-05
ROW     5
        0.00000E+00   1.26500E-05
ROW     6
        0.00000E+00   4.67600E-05
ROW     7
        0.00000E+00   4.91800E-05
ROW     8
        0.00000E+00   3.31100E-05
ROW     9
        0.00000E+00   2.47100E-05
ROW     10
        0.00000E+00   1.81400E-05
ROW     11
        0.00000E+00   1.36900E-05
ROW     12
        0.00000E+00   9.83600E-06
ROW     13
        0.00000E+00   6.92000E-06
ROW     14
        0.00000E+00   5.19000E-06

ROW     15
        0.00000E+00   3.60800E-06
ROW     16
        0.00000E+00   2.22400E-06
ROW     17
        0.00000E+00   1.28500E-06
ROW     18
        0.00000E+00   4.94300E-07
ROW     19
        0.00000E+00   9.88600E-08
ROW     20      NULL ROW
27 WP        - (REAL) 1ST DERIVATIVE OF OP
          GENERAL MATRIX

ROW     1
        3.78780E-03   0.00000E+00
ROW     2
        3.78780E-03   6.68800E-04
ROW     3
        3.78780E-03   1.64000E-03
ROW     4
        3.78780E-03   1.74600E-03
```

```
ROW     5
        3.78780E-03  1.74600E-03
ROW     6
        3.78780E-03  2.04600E-03
ROW     7
        3.78780E-03  2.99600E-03
ROW     8
        3.78780E-03  3.77800E-03
ROW     9
        3.78780E-03  4.35100E-03
ROW     10
        3.78780E-03  4.71700E-03
ROW     11
        3.78780E-03  4.90700E-03
ROW     12
        3.78780E-03  5.06200E-03
ROW     13
        3.78780E-03  5.18300E-03
ROW     14
        3.78780E-03  5.26300E-03
ROW     15
        3.78780E-03  5.32400E-03

ROW     16
        3.78780E-03  5.36200E-03
ROW     17
        3.78780E-03  5.38500E-03
ROW     18
        3.78780E-03  5.39800E-03
ROW     19
        3.78780E-03  5.40100E-03
ROW     20
        3.78780E-03  5.40100E-03
28 W        - (REAL) OUTPLANE MODE SHAPES
         GENERAL MATRIX

ROW     1       NULL ROW
ROW     2
        5.68100E-02  5.01600E-03
ROW     3
        1.70450E-01  3.96500E-02
ROW     4
        2.00000E-01  5.28400E-02
ROW     5
        2.00100E-01  5.28500E-02
```

```
    ROW     6
            2.38200E-01   7.20000E-02
    ROW     7
            3.13250E-01   1.21900E-01
    ROW     8
            3.85200E-01   1.86300E-01
    ROW     9
            4.60200E-01   2.66800E-01
    ROW    10
            5.25000E-01   3.44300E-01
    ROW    11
            5.70000E-01   4.01600E-01
    ROW    12
            6.20000E-01   4.67300E-01
    ROW    13
            6.75000E-01   5.41600E-01
    ROW    14
            7.25000E-01   6.10600E-01
    ROW    15
            7.77000E-01   6.83400E-01
    ROW    16
            8.27100E-01   7.54000E-01

    ROW    17
            8.75900E-01   8.23200E-01
    ROW    18
            9.30000E-01   9.00000E-01
    ROW    19
            9.75000E-01   9.64000E-01
    ROW    20
            1.00000E+00   1.00000E+00
29 PHPP       - (REAL) 2ND DERIVATIVE OF TO
             NULL MATRIX

30 PHP        - (REAL) 1ST DERIVATIVE OF TO
             GENERAL MATRIX

    ROW     1
            0.00000E+00   3.78780E-03
    ROW     2
            0.00000E+00   3.78780E-03
    ROW     3
            0.00000E+00   3.78780E-03
    ROW     4
            0.00000E+00   3.78780E-03
```

```
ROW     5
        0.00000E+00   3.78780E-03
ROW     6
        0.00000E+00   3.78780E-03
ROW     7
        0.00000E+00   3.78780E-03
ROW     8
        0.00000E+00   3.78780E-03
ROW     9
        0.00000E+00   3.78780E-03
ROW     10
        0.00000E+00   3.78780E-03
ROW     11
        0.00000E+00   3.78780E-03
ROW     12
        0.00000E+00   3.78780E-03
ROW     13
        0.00000E+00   3.78780E-03
ROW     14
        0.00000E+00   3.78780E-03
ROW     15
        0.00000E+00   3.78780E-03

ROW     16
        0.00000E+00   3.78780E-03
ROW     17
        0.00000E+00   3.78780E-03
ROW     18
        0.00000E+00   3.78780E-03
ROW     19
        0.00000E+00   3.78780E-03
ROW     20
        0.00000E+00   3.78780E-03
31 PH        - (REAL) TORSION MODE SHAPES
          GENERAL MATRIX

ROW     1
        1.00000E+00   0.00000E+00
ROW     2
        1.00000E+00   5.68100E-02
ROW     3
        1.00000E+00   1.70450E-01
ROW     4
        1.00000E+00   2.00000E-01
ROW     5
```

```
            1.00000E+00   2.00100E-01
    ROW     6
            1.00000E+00   2.38300E-01
    ROW     7
            1.00000E+00   3.13250E-01
    ROW     8
            1.00000E+00   3.85200E-01
    ROW     9
            1.00000E+00   4.60200E-01
    ROW     10
            1.00000E+00   5.25000E-01
    ROW     11
            1.00000E+00   5.70000E-01
    ROW     12
            1.00000E+00   6.20000E-01
    ROW     13
            1.00000E+00   6.75000E-01
    ROW     14
            1.00000E+00   7.25000E-01
    ROW     15
            1.00000E+00   7.77000E-01
    ROW     16

            1.00000E+00   8.27000E-01
    ROW     17
            1.00000E+00   8.76000E-01
    ROW     18
            1.00000E+00   9.30000E-01
    ROW     19
            1.00000E+00   9.75000E-01
    ROW     20
            1.00000E+00   1.00000E+00
32 CIPP      - IP MODAL DAMPING      =  0.00000E+00
33 COPP      - OP MODAL DAMPING      =  0.00000E+00   0.00000E+00
34 CTORR     - TORSION MODAL         =  0.00000E+00   0.00000E+00
35 NI        - NO. OF IMPLICIT DOFS=             0
36 KIP       - IP SPRING RATE        =  0.00000E+00
37 CIP       - IP DAMPING RATE       =  0.00000E+00
38 KOP       - OP SPRING RATE        =  0.00000E+00
39 COP       - OP DAMPING RATE       =  0.00000E+00
40 KTOR      - TORSION SPRING RATE =    0.00000E+00
41 CTOR      - TORSION DAMPING RATE=    0.00000E+00
42 OM        - RPM                   =  3.24000E+02
43 IC        - ROTATION DIRECTION    =             1
44 PSIO      - AZIMUTH OF REF BLADE=    0.00000E+00
```

```
45 BPCO        - PRECONE ANGLE (DEG) =  0.00000E+00
46 MHUB        - HUB WEIGHT (LB)     =  0.00000E+00
47 IHUBX       - HUB M.O.I. ABOUT X- =  0.00000E+00
48 IHUBY       - HUB M.O.I. ABOUT Y- =  0.00000E+00
49 THO         - ROOT PTCH ANG (DEG) =  1.50000E+01
50 NONLIN      - NONLIN TERMS        =          NO
51 IU          - UNIFORM BLADE       =          NO
52 M           - (REAL) MASS PER UNIT LENGTH
               5.78340E+00  5.78340E+00  5.26000E+00  5.26000E+00
               8.90000E-01  8.90000E-01  8.20000E-01  8.44600E-01
               7.97000E-01  7.26000E-01  8.75000E-01  1.09800E+00
               1.06300E+00  1.03900E+00  1.26000E+00  1.18600E+00
               1.26600E+00  1.18900E+00  1.16000E+00  1.16000E+00
53 SE          - (REAL) CG OFFSET FROM EA
               0.00000E+00  0.00000E+00  3.60000E+00  3.60000E+00
               2.50000E+00  2.50000E+00  1.60000E+00  1.60000E+00
               1.60000E+00  1.60000E+00  1.34000E+00  8.50000E-01
               5.71000E-01 -5.50000E-01  1.00000E-02 -1.00000E+00
              -1.87000E+00 -1.12900E+00 -1.97000E+00 -1.97000E+00
54 SEA         - (REAL) AREA CENTROID OFFSET
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
55 KM1         - (REAL) MASS ROG ABOUT
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
56 KM2         - (REAL) MASS ROG ABOUT
               2.75000E+00  2.75000E+00  3.47000E+00  3.47000E+00
               7.10000E+00  7.10000E+00  7.78000E+00  8.05000E+00
               8.16500E+00  8.26500E+00  7.48000E+00  7.20000E+00
               6.90000E+00  6.60000E+00  6.13500E+00  5.60000E+00
               6.00000E+00  5.84000E+00  7.11000E+00  7.11000E+00
57 KA          - (REAL) AREA ROG OF CROSS
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
               0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
58 THP         - (REAL) BUILT-IN TWIST (DEG)
```

```
               -3.78753E-02 -3.78753E-02 -3.78753E-02 -3.78753E-02
               -3.78753E-02 -3.78753E-02 -3.78753E-02 -3.78753E-02
               -3.78753E-02 -3.78753E-02 -3.78753E-02 -3.78753E-02
               -3.78753E-02 -3.78753E-02 -3.78753E-02 -3.78753E-02
               -3.78753E-02 -3.78753E-02 -3.78753E-02 -3.78753E-02
  59 EIY       - (REAL) CHORDWISE EI*10E-6
                5.00000E+03  5.00000E+03  5.00000E+03  5.00000E+03
                4.24000E+03  4.24000E+03  4.15000E+03  4.29000E+03
                3.82000E+03  3.60000E+03  3.39000E+03  3.25500E+03
                2.91000E+03  2.65000E+03  2.65000E+03  2.65000E+03
                2.65500E+03  2.66000E+03  2.69000E+03  2.69000E+03
  60 EIZ       - (REAL) BEAMWISE EI*10E-6
                1.80000E+02  1.80000E+02  3.00000E+02  3.00000E+02
                8.95000E+01  8.95000E+01  5.80000E+01  5.30000E+01
                4.60000E+01  4.00000E+01  4.00000E+01  4.10000E+01
                4.10000E+01  3.96000E+01  3.95000E+01  4.20000E+01
                4.24000E+01  4.24000E+01  3.80000E+01  3.80000E+01
  61 EA        - (REAL) SECTION EA*10E-6
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
  62 GJ        - (REAL) SECTION GJ*10E-6
                3.60000E+01  3.60000E+01  6.70000E+01  6.70000E+01
                7.40000E+01  7.40000E+01  5.70000E+01  5.10000E+01
                4.00000E+01  3.40000E+01  3.38000E+01  3.38000E+01
                3.38000E+01  3.38000E+01  3.38000E+01  3.38000E+01
                3.38000E+01  3.38000E+01  3.38000E+01  3.38000E+01
  63 EB1       - (REAL) CROSS SEC INTEGRAL
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
  64 EB2       - (REAL) CROSS SEC INTEGRAL
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
  65 EC1       - (REAL) CROSS SEC INTEGRAL
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
```

```
                 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 66 EC1STA     - (REAL) CROSS SEC INTEGRAL
                 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
                 0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 67 JIL        - INTERNAL LOADS      =         NO
*****************************************************************


LIST COMPLETE
COMMAND
```

```
LIST
DATA SET
FCT1.65
DATA MEMBER
FRA3
FCT1.65 /FRA3      ON FILE U1


***************        FCT1.65 /FRA3       ***************

GENERAL AERO, INDUCED VEL 1.65

REQUIRES DS/DM AFD161  /AIRFOIL
NO SEQUENTIAL FILES REQUIRED

*****************************************************************
INPUT FOR FORCE FRA3. ROTOR AERO GENERAL

  1 IEQS      - AERODYNAMICS BY EQS =         NO
  2 INFTAB    - INDUCED VEL BY TABLE=         NO
  3 IUNSTD    - UNSTEADY AERO       =         NO
  4 VAIRH     - (REAL) WIND VELOCITY

              0.00000E+00  0.00000E+00  0.00000E+00
  5 ASTALL    - STALL ANGLE (DEG)   =  2.00000E+01
  6 RFCT      - INDUCED VEL FACTOR  =  1.65000E+00
  7 TIPLOC    - TIP LOSS COEFFICIENT=  9.50000E-01
  8 XH        - HUB EXTENT (IN)     =  3.96000E+00
  9 ALT       - VEHICLE HEIGHT (FT) =  2.00000E+02
 10 K27       - TIP VORTEX COEFF    =  0.00000E+00
 11 CD0       - BLADE DRAG COEFFAT  =  6.80000E-03
 12 Q1C       - Q1C COEFFICIENT     =  1.00000E+00
 13 Q2C       - Q2C COEFFICIENT     =  5.00000E-01
 14 ALAMDA    - NONDIM INDUCED VEL  = -1.20000E-02
 15 NXA       - NO. OF STATIONS     =           12
 16 XAERO     - (REAL) NONDIM AERO STATIONS
              1.77000E-01  2.25000E-01  4.00000E-01  6.00000E-01
              7.00000E-01  7.50000E-01  8.00000E-01  8.50000E-01
              9.00000E-01  9.35000E-01  9.70000E-01  1.00000E+00
 17 NUMAF     - NO. AIRFOIL TABLES  =            1
 18 AFTAB1    - NAME AF    TABLE 1  = AFD161   /AIRFOIL
 19 NUMAF1    - NO. OF STATIONS AF 1=           12
 20 STA-AF1   - STATIONS FOR AF 1
                     1          2          3          4          5
                     6          7          8          9         10
```

```
                   11           12
 21 XACC      - (REAL) A/C OFFSET FROM E.C.
              0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
              0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
              0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
 22 CHORDC    - (REAL) CHORD (IN)
              2.85000E+01  2.85000E+01  2.85000E+01  2.85000E+01
              2.85000E+01  2.85000E+01  2.85000E+01  2.85000E+01
              2.85000E+01  2.85000E+01  2.85000E+01  2.85000E+01
 23 NX        - NO. AERO FACTOR STAS=                2
 24 XF        - NONDIM FACTOR STAS  =  0.00000E+00  1.00000E+00
 25 FL        - FACTORS FOR CL      =  1.00000E+00  1.00000E+00
 26 FD        - FACTORS FOR CD      =  1.00000E+00  1.00000E+00
 27 FM        - FACTORS FOR CM      =  1.00000E+00  1.00000E+00
****************************************************************


LIST COMPLETE
COMMAND
```

```
LIST
DATA SET
3000
DATA MEMBER
CCE0
3000    /CCE0     ON FILE U1


***************       3000    /CCE0       ***************

CONTROL ROD STIFFNESS 3000 LB/IN

***********************************************************
INPUT FOR CONTROL SYSTEM COMPONENT CCE0. CONTROL RODS

  1 KROD     - CONTRL ROD STIFFNESS=  3.00000E+03
***********************************************************



LIST COMPLETE
COMMAND
```

```
LIST
DATA SET
COUPLE
DATA MEMBER
CLC1
COUPLE  /CLC1     ON FILE U1


***************      COUPLE  /CLC1      ***************

TEETERING CONSTRAINT

**********************************************************************
INPUT FOR COMPONENT CLC1. LINEAR CONSTRAINTS

  1 NCDF     - NUMBER OF DOF       =           5
  2 CDFLI    - (DOF) DOF NAMES
           TEET   0  OPOP1120  OPOP1220  IPIP1110  IPIP1210
  3 NCIDF    - # OF CONSTRAINT EQNS=           6
  4 CIDFLI   - (DOF) IMPLICIT DOF NAMES
           OP  1110  OP  1210  OP  1120  OP  1220  IP  1110
           IP  1210

  5 COEF     - (REAL) COEFFICIENT MATRIX
           GENERAL MATRIX

    ROW    1
          2.64000E+02  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
    ROW    2
         -2.64000E+02  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
    ROW    3
          0.00000E+00  2.64000E+02  0.00000E+00  0.00000E+00
          0.00000E+00
    ROW    4
          0.00000E+00  0.00000E+00  2.64000E+02  0.00000E+00
          0.00000E+00
    ROW    5
          0.00000E+00  0.00000E+00  0.00000E+00  2.64000E+02
          0.00000E+00
    ROW    6
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          2.64000E+02
**********************************************************************


LIST COMPLETE
COMMAND
```

```
LIST
DATA SET
8300-4
DATA MEMBER
CFM2
8300-4   /CFM2      ON FILE U1


***************       8300-4   /CFM2       ***************

8300 LB AH1G, HUB (-.68, 96.485)

*****************************************************************
INPUT FOR STRUCTURAL     COMPONENT CFM2. MODAL FUSELAGE

   1 RBM       - RIGID BODY MODES     =          YES
   2 IXCG      - LONGITUDINAL         =          YES
   3 IYCG      - LATERAL              =          NO
   4 IZCG      - VERTICAL             =          YES
   5 IROLL     - ROLL                 =          YES
   6 IPTCH     - PITCH                =          YES
   7 IYAW      - YAW                  =          NO

   8 CG        - CG STATION (IN)      =   0.00000E+00
   9 NMODE     - NO. OF ELASTIC MODES=              0
  10 NR        - NO. OF ROTORS        =              1
  11 NROT      - ROTOR NUMBERS        =              1
  12 XROT      - ROTOR STATIONS       =  -6.80000E-01
  13 ZROT      - ROTOR VERTICAL HT    =   9.64850E+01
  14 ASF       - FWD SHAFT ANGLE      =   0.00000E+00
  15 ASL       - LAT SHAFT ANGLE      =   0.00000E+00
  16 IX        - HUB TRAN DOF - LONG =           YES
  17 IY        - HUB TRAN DOF - LAT  =           NO
  18 IZ        - HUB TRAN DOF - AXIAL=           YES
  19 IAX       - HUB ANGL DOF - ROLL =           YES
  20 IAY       - HUB ANGL DOF - PITCH=           YES
  21 IAZ       - HUB ANGL DOF - YAW  =           NO
  22 NI        - NO. OTHER IMPLCT DOF=             0
  23 MASSL     - FUSELAGE MASS  (LB) =   7.29140E+03
  24 IMXF      - ROLL MOI SLUG-FT(SQ)=   3.00000E+03
  25 IMYF      - PITCH MOI ABOUT CG  =   8.00000E+03
*****************************************************************


LIST COMPLETE
COMMAND
```

```
LIST
DATA SET
AH1G16.5
DATA MEMBER
FFC2
AH1G16.5/FFC2     ON FILE U1


***************      AH1G16.5/FFC2      ***************

AH1G, 16.5 SQ FT FLAT PLATE DRAG

NO AUXILIARY DS/DM(S) REQUIRED
NO SEQUENTIAL FILES REQUIRED

******************************************************************
INPUT FOR FORCE FFC2. FUSELAGE AERO LINEAR

  1 VAIR     - (REAL) WIND VELOCITY
             0.00000E+00  0.00000E+00  0.00000E+00
  2 IFUSE    - FUSELAGE AERO FORCES=          YES
  3 RFUSE    - (REAL) FUSELAGE A/C LOC

            -6.80000E-01  0.00000E+00  1.49250E+01
  4 AFUSE    - FUSELAGE AREA        =  1.00000E+00
  5 AL0FUS   - FUSLAG LIFT COEFF C0=  0.00000E+00
  6 AL1FUS   - FUSLAG LIFT COEFF C1=  3.20000E-01
  7 AL2FUS   - FUSLAG LIFT COEFF C2=  2.50000E-03
  8 CD0FUS   - FUSLAG DRAG COEFF C0=  1.65000E+01
  9 CD1FUS   - FUSLAG DRAG COEFF C1=  3.20000E-02
 10 CD2FUS   - FUSLAG DRAG COEFF C2=  1.00000E-02
 11 CM0FUS   - FUSELAG MOM COEFF C0= -3.24000E+02
 12 CM1FUS   - FUSELAG MOM COEFF C1=  4.80000E+01
 13 CM2FUS   - FUSELAG MOM COEFF C2= -1.20000E-01
 14 FUINCI   - FUSELAGE INCIDENCE  =  0.00000E+00
 15 FUDO     - FUSLAG DOWNWASH ANGL=  0.00000E+00
 16 FURATI   - FUSLAG VELOCTY RATIO=  1.00000E+00
 17 IWING    - CONSIDER WING       =          YES
 18 RWING    - (REAL) WING A/C LOC
            -1.09500E+01  0.00000E+00  1.20050E+01
 19 AWING    - WING AREA (SQR FT)  =  2.81500E+01
 20 AL0WIN   - WING LIFT COEFF C0  =  5.80000E-01
 21 AL1WIN   - WING LIFT COEFF C1  =  6.00000E-02
 22 AL2WIN   - WING LIFT COEFF C2  = -7.00000E-03
 23 CD0WIN   - WING DRAG COEFF C0  =  6.40000E-02
```

```
 24 CD1WIN    - WING DRAG COEFF C1   =  0.00000E+00
 25 CD2WIN    - WING DRAG COEFF C2   =  0.00000E+00
 26 CMOWIN    - WING MOM COEFF C0    = -1.00000E+00
 27 CM1WIN    - WING MOM COEFF C1    =  0.00000E+00
 28 CM2WIN    - WING MOM COEFF C2    =  0.00000E+00
 29 WINCI     - WING INCIDENCE (DEG)=  4.80000E+00
 30 WINGDO    - WING DOWNWASH ANGLE =  0.00000E+00
 31 WRATI     - WING VELOCITY RATIO =  1.00000E+00
 32 IHT       - CONSIDER HORIZ TAIL =          YES
 33 RHT       - (REAL) HTAIL A/C LOC
              1.96820E+02  0.00000E+00 -4.08000E+01
 34 AHT       - HORIZONTAL TAIL AREA=  1.52000E+01
 35 ALOHT     - HTAIL LIFT COEFF C0 = -2.50000E-01
 36 AL1HT     - HTAIL LIFT COEFF C1 =  7.55000E-03
 37 AL2HT     - HTAIL LIFT COEFF C2 =  1.52000E-03
 38 CDOHT     - HTAIL DRAG COEFF C0 =  2.50000E-02
 39 CD1HT     - HTAIL DRAG COEFF C1 =  0.00000E+00
 40 CD2HT     - HTAIL DRAG COEFF C2 =  0.00000E+00
 41 CMOHT     - HTAIL MOM COEFF C0  =  0.00000E+00
 42 CM1HT     - HTAIL MOM COEFF C1  =  0.00000E+00
 43 CM2HT     - HTAIL MOM COEFF C2  =  0.00000E+00
 44 HTINCI    - HTAIL INCIDENCE     =  6.87000E+00

 45 HTDO      - HTAIL DOWNWASH ANGLE=  0.00000E+00
 46 HTRATI    - HTAIL VELOCITY RATIO=  1.00000E+00
 47 IVT       - CONSIDER VERT TAIL  =          YES
 48 RVT       - (REAL) VTAIL A/C LOC
              3.00320E+02  0.00000E+00  7.00000E-01
 49 AVT       - VERTICAL TAIL AREA  =  1.88700E+01
 50 ALOVT     - VTAIL LIFT COEFF C0 =  2.20000E-01
 51 AL1VT     - VTAIL LIFT COEFF C1 =  0.00000E+00
 52 AL2VT     - VTAIL LIFT COEFF C2 =  0.00000E+00
 53 CDOVT     - VTAIL DRAG COEFF C0 =  5.00000E-02
 54 CD1VT     - VTAIL DRAG COEFF C1 =  0.00000E+00
 55 CD2VT     - VTAIL DRAG COEFF C2 =  0.00000E+00
 56 CMOVT     - VTAIL MOM COEFF C0  =  0.00000E+00
 57 CM1VT     - VTAIL MOM COEFF C1  =  0.00000E+00
 58 CM2VT     - VTAIL MOM COEFF C2  =  0.00000E+00
 59 VTINCI    - VTAIL INCIDENCE     =  4.50000E+00
 60 VTDO      - VTAIL DOWNWASH ANGLE=  0.00000E+00
 61 VTRATI    - VTAIL VELOCITY RATIO=  1.00000E+00
 62 ITAIL     - CONSIDER TAIL ROTOR =          NO
 63 IPROP     - CONSIDER PROPELLER  =          NO
*******************************************************************


LIST COMPLETE
COMMAND
```

```
EXECUTION BEGINS...
USER FILE U1 TO BE INITIALIZED (Y OR N)
COMMAND
MODEL NAME (DATA SET)
LIST MODEL SUMMARY (Y OR N)


***********************      MODEL AH1G-35A  ******************

AH1G TRIM

INDEX    COMP    NO.    DATA SET         FORCE    DATA SET

  1      CRE3     1     B2Z1T2           FRA3     FCT1.65
                                    REQUIRED DS/DM= AFD161   /AIRFOIL
  2      CCE0     1     3000             NONE
  3      CLC1           COUPLE           NONE
  4      CFM2     1     8300-4           FFC2     AH1G16.5

***************************************************************


***************************************************************
                                 GLOBAL  VARIABLES

  1 VSOUND    - SOUND VELOCITY       =  1.13800E+03
  2 RHO       - AIR DENSITY RATIO    =  8.79000E-01
***************************************************************


TEMPORARY RUN EDIT OF GLOBAL VARIABLES FOR MODEL (Y OR N)
TEMPORARY RUN EDIT OF ANY COMPONENT/FORCE INPUT (Y OR N)
***************************************************************
DETAILS (Y OR N)


 **************************************************************
                          COMPONENT DOF/SYSTEM DOF
                   1, 6         2, 7         3, 8         4, 9         5, 10


  1  CRE3     IP  1110     OP  1110     OP  1120     TOR 1110     TOR 112
```

		IP 1210	OP 1210	OP 1220	TOR 1210	TOR 122
		XHUB1000	ZHUB1000	ALFX1000	ALFY1000	
		(-1)	(-2)	(-3)	(1)	(2)
		(-4)	(-5)	(-6)	(3)	(4)
		(-7)	(-9)	(-11)	(-12)	
2	CCEO	RODR1100	RODR1200			
		(-13)	(-17)			
3	CLC1	TEET 0	OPOP1120	OPOP1220	IPIP1110	IPIP121
		(5)	(6)	(7)	(8)	(9)
4	CFM2	XCG 1000	ZCG 1000	ROLL1000	PTCH1000	
		(10)	(11)	(12)	(13)	

SYSTEM DOF

1	TOR 1110
2	TOR 1120
3	TOR 1210
4	TOR 1220
5	TEET 0
6	OPOP1120
7	OPOP1220
8	IPIP1110
9	IPIP1210
10	XCG 1000
11	ZCG 1000
12	ROLL1000
13	PTCH1000

IMPLICIT COEFFICIENTS

I	COEF	DOF	I	COEF	DOF
1	2.640E+02	*IPIP1110	11	1.000E+00	*ROLL1000
2	2.640E+02	*TEET 0	12	1.000E+00	*PTCH1000
3	2.640E+02	*OPOP1120	13	-9.250E+00	TOR 1110
4	2.640E+02	*IPIP1210	14	-4.940E-01	TOR 1120
5	-2.640E+02	*TEET 0	15	1.410E+01	TEET 0
6	2.640E+02	*OPOP1220	16	1.245E+00	*OPOP1120

```
   7    1.000E+00     XCG 1000     17   -9.250E+00      TOR 1210
   8    9.649E+01    *PTCH1000     18   -4.940E-01      TOR 1220
   9    1.000E+00     ZCG 1000     19   -1.410E+01      TEET    0
  10    6.800E-01    *PTCH1000     20    1.245E+00     *OPOP1220
PRINT MATRICES (Y OR N)
MASS (Y OR N)
            GENERAL MATRIX

     ROW     1
           3.90899E+01  1.80789E+01  0.00000E+00  0.00000E+00
           3.56521E+01 -2.00903E+01  0.00000E+00 -3.80490E+00
           0.00000E+00  0.00000E+00  1.26491E+00  0.00000E+00
           8.60137E-01
     ROW     2
           1.80789E+01  1.15905E+01  0.00000E+00  0.00000E+00
          -2.53690E+01 -3.78054E+01  0.00000E+00  2.14633E+00
           0.00000E+00  0.00000E+00  1.35050E-01  0.00000E+00
           9.18339E-02
     ROW     3
           0.00000E+00  0.00000E+00  3.90899E+01  1.80789E+01
          -3.56521E+01  0.00000E+00 -2.00903E+01  0.00000E+00
          -3.80490E+00  0.00000E+00  1.26491E+00  0.00000E+00

           8.60137E-01
     ROW     4
           0.00000E+00  0.00000E+00  1.80789E+01  1.15905E+01
           2.53690E+01  0.00000E+00 -3.78054E+01  0.00000E+00
           2.14633E+00  0.00000E+00  1.35050E-01  0.00000E+00
           9.18339E-02
     ROW     5
           3.56521E+01 -2.53690E+01 -3.56521E+01  2.53690E+01
           3.60922E+04  1.50029E+04 -1.50029E+04  0.00000E+00
           0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
           0.00000E+00
     ROW     6
          -2.00903E+01 -3.78054E+01  0.00000E+00  0.00000E+00
           1.50029E+04  1.31302E+04  0.00000E+00  0.00000E+00
           0.00000E+00  0.00000E+00  7.72289E+01  0.00000E+00
           5.25157E+01
     ROW     7
           0.00000E+00  0.00000E+00 -2.00903E+01 -3.78054E+01
          -1.50029E+04  0.00000E+00  1.31302E+04  0.00000E+00
           0.00000E+00  0.00000E+00  7.72289E+01  0.00000E+00
           5.25157E+01
     ROW     8
```

```
        -3.80490E+00  2.14633E+00  0.00000E+00  0.00000E+00
         0.00000E+00  0.00000E+00  0.00000E+00  1.40185E+04
         0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         0.00000E+00
   ROW    9
         0.00000E+00  0.00000E+00 -3.80490E+00  2.14633E+00
         0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         1.40185E+04  0.00000E+00  0.00000E+00  0.00000E+00
         0.00000E+00
   ROW   10
         0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         0.00000E+00  2.14756E+01  0.00000E+00  0.00000E+00
         2.52021E+02
   ROW   11
         1.26491E+00  1.35050E-01  1.26491E+00  1.35050E-01
         0.00000E+00  7.72289E+01  7.72289E+01  0.00000E+00
         0.00000E+00  0.00000E+00  2.14756E+01  0.00000E+00
         1.77176E+00
   ROW   12
         0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

         0.00000E+00  0.00000E+00  0.00000E+00  5.40469E+04
         0.00000E+00
   ROW   13
         8.60137E-01  9.18339E-02  8.60137E-01  9.18339E-02
         0.00000E+00  5.25157E+01  5.25157E+01  0.00000E+00
         0.00000E+00  2.52021E+02  1.77176E+00  0.00000E+00
         1.38425E+05
DAMPING (Y OR N)
          GENERAL MATRIX

   ROW    1       NULL ROW
   ROW    2       NULL ROW
   ROW    3       NULL ROW
   ROW    4       NULL ROW
   ROW    5
         0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         0.00000E+00  0.00000E+00  0.00000E+00  2.58189E+02
        -2.58189E+02  0.00000E+00  0.00000E+00  0.00000E+00
         0.00000E+00
   ROW    6
         0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         0.00000E+00  0.00000E+00  0.00000E+00 -5.02048E+01
```

```
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
     ROW     7
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         -5.02048E+01  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
     ROW     8
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
         -2.58189E+02  5.02048E+01  0.00000E+00 -1.10116E-03
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
     ROW     9
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          2.58189E+02  0.00000E+00  5.02048E+01  0.00000E+00
         -1.10116E-03  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
     ROW    10      NULL ROW
     ROW    11      NULL ROW
     ROW    12
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          1.22463E+06
     ROW    13
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00  0.00000E+00  0.00000E+00 -1.22463E+06
          0.00000E+00
STIFFNESS (Y OR N)
            GENERAL MATRIX

     ROW     1
          2.95075E+05  3.27416E+04  0.00000E+00  0.00000E+00
         -3.48058E+05 -5.13820E+04  0.00000E+00 -5.31666E+03
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
     ROW     2
          3.27416E+04  1.77589E+05  0.00000E+00  0.00000E+00
         -4.97757E+04 -6.00735E+04  0.00000E+00  6.19839E+01
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
     ROW     3
          0.00000E+00  0.00000E+00  2.95075E+05  3.27416E+04
```

```
          3.48058E+05  0.00000E+00 -5.13820E+04  0.00000E+00
         -5.31666E+03  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
    ROW    4
          0.00000E+00  0.00000E+00  3.27416E+04  1.77589E+05
          4.97757E+04  0.00000E+00 -6.00735E+04  0.00000E+00
          6.19839E+01  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
    ROW    5
         -3.48058E+05 -4.97757E+04  3.48058E+05  4.97757E+04
          4.24301E+07  1.71957E+07 -1.71957E+07  0.00000E+00
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
    ROW    6
         -5.13820E+04 -6.00735E+04  0.00000E+00  0.00000E+00
          1.71957E+07  2.31974E+07  0.00000E+00  1.34004E+07
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
    ROW    7
          0.00000E+00  0.00000E+00 -5.13820E+04 -6.00735E+04
         -1.71957E+07  0.00000E+00  2.31974E+07  0.00000E+00
          1.34004E+07  0.00000E+00  0.00000E+00  0.00000E+00

          0.00000E+00
    ROW    8
         -5.31666E+03  6.19839E+01  0.00000E+00  0.00000E+00
          0.00000E+00  1.34004E+07  0.00000E+00  8.10297E+07
          0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
    ROW    9
          0.00000E+00  0.00000E+00 -5.31666E+03  6.19839E+01
          0.00000E+00  0.00000E+00  1.34004E+07  0.00000E+00
          8.10297E+07  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
    ROW   10       NULL ROW
    ROW   11       NULL ROW
    ROW   12       NULL ROW
    ROW   13       NULL ROW
FORCE (Y OR N)
         -6.68374E+03 -2.77793E+03 -6.68374E+03 -2.77793E+03
          0.00000E+00 -5.66349E+03 -5.66349E+03 -1.23520E+04
         -1.23520E+04  0.00000E+00  0.00000E+00  0.00000E+00
          0.00000E+00
***************************************************************
SOLUTION OR N
```

```
SAVE CASE FOR LATER EXECUTION (Y OR N)

SOLUTION STR3. TRIM SOLUTION

BEGIN INPUT
-----
IFUS      (Y OR N)
    PERIODIC MOTION
    OF FUSELAGE DOF TO BE CONSIDERED (Y OR N)
ENTER 1 Y OR N VALUE
-----
ITEET     (Y OR N)
    TEETERING ROTOR
    (Y OR N)
ENTER 1 Y OR N VALUE
-----
TRCOVA7   (REAL)
    VALUE A
    GIVE A,B,C VALUES SO THAT THE CYCLIC SINE CONTROL RELATES TO THE
    HORIZONTAL TAIL INCIDENCE IN THE FOLLOWING WAY:
    INCIDENCE ANGLE = A + B*A1S + (C*A1S)**2


ENTER 1 REAL VALUE
-----
TRCOVA8   (REAL)
    VALUE B
ENTER 1 REAL VALUE
-----
TRCOVA9   (REAL)
    VALUE C
ENTER 1 REAL VALUE
-----
H         (REAL)
    INITIAL INCREMENT
    (SEC)
ENTER 1 REAL VALUE
-----
TPER      (REAL)
    INTEGRATION PERIOD
ENTER 1 REAL VALUE
-----
HTD       (REAL)
    SEPARATE INCREMENT
```

```
    TIME DEP COEFS
ENTER 1 REAL VALUE
-----
HF        (REAL)
    SEPARATE INCREMENT
    FORCE COMPUTATION
ENTER 1 REAL VALUE
-----
E         (REAL)
    ERROR CHECK VALUE
    IF 0 THEN CONSTANT INCREMENT USED
ENTER 1 REAL VALUE
-----
ITRIM     (INTEGER)
    CASE NUMBER
    CASE 1 PERFORM FORCE BALANCE IN XCG,ZCG,ALFX,ALFY
           TRIM PARAMETERS: A0,A1C,A1S,PTCH
    CASE 2 PERFORM FORCE BALANCE IN XCG,YCG,ZCG,ALFX,ALFY,
           TRIM PARAMETERS: A0,A1C,A1S, AND 2 EULER ANGLES
           (AROLL, APTCH, AYAW)
    CASE 3 PERFORM FORCE BALANCE IN XCG,YCG,ZCG,ALFX,ALFY,ALFZ
           TRIM PARAMETERS: A0,A1C,A1S,2 EULER ANGLES AND YAW MOMENT

    CASE 4 PERFORM FORCE BALANCE IN XCG,YCG,ZCG,ALFX,ALFY,ALFZ,
           TRIM PARAMETERS: A0,A1C,A1S,2 EULER ANGLES,VEL VEHICLE
ENTER    1 INTEGER VALUE(S)
-----
NALLOW   (INTEGER)
    NO. OF ITERATIONS
    ALLOWED
ENTER    1 INTEGER VALUE(S)
-----
CEA       (Y OR N)
    CONSTANT ERROR
    ALLOWED, ENTER YES. IF NOT ENTER NO
ENTER 1 Y OR N VALUE
-----
CEALLO   (REAL)
    CONSTANT ERROR
ENTER 1 REAL VALUE
-----
EALLOWT1 (REAL)
    TRIM ERRORS ALLOWED
    UNBALANCE FORCES AND MOMNTS CASE 1
NULL VECTOR (Y OR N)
```

```
ENTER  4 REAL VALUE(S)
-----
CI        (Y OR N)
    CONSTANT INCREMENT
    FOR COMPUTING TRANSITION MATRIX, ENTER YES ELSE ENTER NO
ENTER 1 Y OR N VALUE
-----
CICRE     (REAL)
    CONSTANT INCREMENT
    TO BE USED
ENTER 1 REAL VALUE
-----
DELTAT1   (REAL)
    CONTROL VAR INCRMNTS
    INPUT TRANSITION MATRX INCRMNT FOR EACH VARIABLE
NULL VECTOR (Y OR N)
ENTER  4 REAL VALUE(S)
-----
IWIND     (Y OR N)
    WIND VELOCITY Y OR N
ENTER 1 Y OR N VALUE
-----

VTRANG    (REAL)
    FUSELAGE TRANS VEL
    TRANSLATIONAL VELOCITY IN TERMS OF GROUND FIXED COORD X,Y,Z
NULL VECTOR (Y OR N)
ENTER  3 REAL VALUE(S)
-----
IROTIN    (Y OR N)
    FUSELAGE ANGULAR VEL
    Y OR N
ENTER 1 Y OR N VALUE
-----
CT        (REAL)
    ROTOR THRUST
    ESTIMATE
ENTER 1 REAL VALUE
-----
IGUESS    (Y OR N)
    INITIAL GUESSES
    FOR CONTROLS Y OR N
ENTER 1 Y OR N VALUE
-----
AFTCH     (REAL)
```

```
    GUESS FUSELAGE ANGLE
    APTCH
ENTER 1 REAL VALUE
-----
ISTEAD   (Y OR N)
    STEADY TIME HISTORY
    TO BE OUTPUT AFTER TRIM SOLUTION (Y OR N)
ENTER 1 Y OR N VALUE
-----
CRT      (Y OR N)
    OUTPUT THIS TERMINAL
    Y OR N
ENTER 1 Y OR N VALUE
******************************************************************
SOLUTION INPUT FOR STR3.TRIM SOLUTION

  1 IFUS      - PERIODIC MOTION      =          NO
  2 ITEET     - TEETERING ROTOR      =         YES
  3 TRCOVA7   - VALUE A              =  1.00000E-01
  4 TRCOVA8   - VALUE B              = -1.24000E-01
  5 TRCOVA9   - VALUE C              =  3.52000E+00
  6 H         - INITIAL INCREMENT    =  3.70370E-03

  7 TPER      - INTEGRATION PERIOD   =  1.85185E-01
  8 HTD       - SEPARATE INCREMENT   =  0.00000E+00
  9 HF        - SEPARATE INCREMENT   =  0.00000E+00
 10 E         - ERROR CHECK VALUE    =  0.00000E+00
 11 ITRIM     - CASE NUMBER          =            1
 12 NALLOW    - NO. OF ITERATIONS    =            8
 13 CEA       - CONSTANT ERROR       =          YES
 14 CEALLO    - CONSTANT ERROR       =  1.00000E-05
 15 EALLOWT1  - (REAL) TRIM ERRORS ALLOWED
              1.00000E-05  1.00000E-05  1.00000E-05  1.00000E-05
 16 CI        - CONSTANT INCREMENT  =         YES
 17 CICRE     - CONSTANT INCREMENT  =  5.00000E-03
 18 DELTAT1   - (REAL) CONTROL VAR INCRMNTS
              5.00000E-03  5.00000E-03  5.00000E-03  5.00000E-03
 19 IWIND     - WIND VELOCITY Y OR N=          NO
 20 VTRANG    - (REAL) FUSELAGE TRANS VEL
             -1.36800E+03  0.00000E+00  0.00000E+00
 21 IROTIN    - FUSELAGE ANGULAR VEL=          NO
 22 CT        - ROTOR THRUST        =  8.00000E+03
 23 IGUESS    - INITIAL GUESSES     =          NO
 24 APTCH     - GUESS FUSELAGE ANGLE= -1.50000E-02
 25 ISTEAD    - STEADY TIME HISTORY =          NO
```

```
 26 CRT       - OUTPUT THIS TERMINAL=        YES
*****************************************************************

RE-ENTER (Y OR N)

***************** SOLUTION STR3 FOR MODEL AH1G-35A***************
MODEL      - AH1G TRIM
SOLUTION - TRIM SOLUTION


NO. OF ITERATION =    1

INITIAL CONDITIONS ARE:

                   VELOCITY      DISPLACEMENT

  TOR 1110         0.206651        0.032732

  TOR 1120        -0.633596        0.041138

  TOR 1210        -1.941247       -0.135377

  TOR 1220         4.441560       -0.019783

  TEET    0        0.562240        0.041226

  OPOP1120        -0.890341        0.034397

  OPOP1220        -0.200073        0.025057

  IPIP1110         0.048268       -0.002325

  IPIP1210        -0.074703       -0.005199

 A0 =    -0.04955   A1C =     0.01164   A1S =    -0.05129

 FUSELAGE PITCH ANGLE =    -0.0171186

 AERODYNAMIC FORCES ARE:

           TORQ          XFOR         YFOR        ZFOR          HP
     -144170.75        282.76       198.97    11905.85     -741.15
```

MOMX	MOMY
52449.19	231336.12

INDUCED VEL = -0.01200

VEL W.R.T. ROTOR SYSTEM ARE:

-1367.9316	0.0000	13.6798

VEL W.R.T. FUSE SYSTEM ARE:

-1367.9316	0.0000	13.6798

UNBALANCE FORCES ARE:

FX	FZ	MX	MY
428.01	3835.17	52409.59	272898.69

FUSLAGE LIFT,DRAG,MOM,FX,FY,FZ:

-2.48	223.91	-4774.82	223.87	0.00	-4.72

WING LIFT,DRAG,MOM,FX,FY,FZ:

270.91	24.47	-382.35	27.18	0.00	270.65

HTAIL LIFT,DRAG,MOM,FX,FY,FZ:

-35.23	5.16	0.00	4.81	0.00	-35.28

VTAIL LIFT,DRAG,MOM,FX,FY,FZ:

56.39	12.82	0.00	12.82	56.39	0.00

NO. OF ITERATION = 2

INITIAL CONDITIONS ARE:

	VELOCITY	DISPLACEMENT
TOR 1110	-2.380661	0.020308
TOR 1120	-1.403514	0.012690
TOR 1210	2.441963	-0.069770

```
  TOR 1220        0.952898         -0.008948

  TEET    0       0.291817          0.046756

  OPOP1120       -0.219892          0.024917

  OPOP1220        0.403480          0.016205

  IPIP1110       -0.043058         -0.003400

  IPIP1210       -0.110872         -0.001995

A0 =    -0.01072   A1C =    -0.03328   A1S =    -0.08239

FUSELAGE PITCH ANGLE =   -0.0092651

AERODYNAMIC FORCES ARE:

        TORQ          XFOR          YFOR          ZFOR          HP
    -55479.47        -46.39       -184.99       4828.00      -285.21

         MOMX          MOMY

    172650.37      59644.43

INDUCED VEL =    -0.02015

VEL W.R.T. ROTOR SYSTEM ARE:
   -1367.8994      0.0000      16.5779

VEL W.R.T. FUSE SYSTEM ARE:
   -1367.8994      0.0000      16.5779

 UNBALANCE FORCES ARE:

        FX          FZ            MX            MY
      81.78    -3239.98     172610.87     63615.17

FUSLAGE LIFT,DRAG,MOM,FX,FY,FZ:
     -3.00      223.88   -4854.21     223.82       0.00      -5.71

WING LIFT,DRAG,MOM,FX,FY,FZ:
    270.83       24.47    -382.35      27.75       0.00     270.52

HTAIL LIFT,DRAG,MOM,FX,FY,FZ:
```

-31.33	5.16	0.00	4.78	0.00	-31.39

VTAIL LIFT,DRAG,MOM,FX,FY,FZ:

56.39	12.82	0.00	12.82	56.39	0.00

NO. OF ITERATION = 3

INITIAL CONDITIONS ARE:

	VELOCITY	DISPLACEMENT
TOR 1110	-1.865746	0.008288
TOR 1120	0.004530	0.005770
TOR 1210	1.755579	-0.069111
TOR 1220	-0.108180	-0.004024
TEET 0	0.247339	0.017590
OPOP1120	-0.211407	0.028230
OPOP1220	0.249587	0.024710
IPIP1110	0.022821	-0.004468
IPIP1210	-0.068222	-0.003531

A0 = -0.01595 A1C = 0.00819 A1S = -0.06086

FUSELAGE PITCH ANGLE = -0.0148709

AERODYNAMIC FORCES ARE:

TORQ	XFOR	YFOR	ZFOR	HP
-90076.75	-334.30	-156.77	8069.21	-463.07

MOMX	MOMY
29957.90	28507.30

INDUCED VEL = -0.02018

VEL W.R.T. ROTOR SYSTEM ARE:

-1367.9875	0.0000	5.8347

VEL W.R.T. FUSE SYSTEM ARE:

-1367.9875	0.0000	5.8347

UNBALANCE FORCES ARE:

FX	FZ	MX	MY
-142.57	13.29	29918.41	6271.56

FUSLAGE LIFT,DRAG,MOM,FX,FY,FZ:

-1.06	224.01	-4560.16	224.01	0.00	-2.02

WING LIFT,DRAG,MOM,FX,FY,FZ:

270.73	24.47	-382.35	25.62	0.00	270.62

HTAIL LIFT,DRAG,MOM,FX,FY,FZ:

-22.65	5.16	0.00	5.06	0.00	-22.67

VTAIL LIFT,DRAG,MOM,FX,FY,FZ:

56.39	12.82	0.00	12.82	56.39	0.00

NO. OF ITERATION = 4

INITIAL CONDITIONS ARE:

	VELOCITY	DISPLACEMENT
TOR 1110	-1.780149	0.010467
TOR 1120	-0.004885	0.005063
TOR 1210	1.695828	-0.069338
TOR 1220	-0.002418	-0.003482
TEET 0	0.277017	0.018530

OPOP1120	-0.231618	0.028061
OPOP1220	0.289122	0.023995
IPIP1110	0.034385	-0.004436
IPIP1210	-0.088226	-0.003352

A0 = -0.01523 A1C = 0.00785 A1S = -0.06263

FUSELAGE PITCH ANGLE = -0.0151406

AERODYNAMIC FORCES ARE:

TORQ	XFOR	YFOR	ZFOR	HP
-84415.44	-138.36	-165.71	7984.50	-433.96

MOMX	MOMY
-1304.50	2051.74

INDUCED VEL = -0.02034

VEL W.R.T. ROTOR SYSTEM ARE:
-1367.9333 0.0000 13.5032

VEL W.R.T. FUSE SYSTEM ARE:
-1367.9333 0.0000 13.5032

UNBALANCE FORCES ARE:

FX	FZ	MX	MY
8.02	-80.23	-1343.94	-835.48

FUSLAGE LIFT,DRAG,MOM,FX,FY,FZ:
-2.45 223.91 -4769.99 223.87 0.00 -4.66

WING LIFT,DRAG,MOM,FX,FY,FZ:
270.91 24.47 -382.35 27.14 0.00 270.66

HTAIL LIFT,DRAG,MOM,FX,FY,FZ:
-29.10 5.16 0.00 4.87 0.00 -29.15

VTAIL LIFT,DRAG,MOM,FX,FY,FZ:
56.39 12.82 0.00 12.82 56.39 0.00

NO. OF ITERATION = 5

INITIAL CONDITIONS ARE:

	VELOCITY	DISPLACEMENT
TOR 1110	-1.774555	0.009883
TOR 1120	-0.015566	0.005197
TOR 1210	1.687477	-0.069429
TOR 1220	-0.007876	-0.003517
TEET 0	0.281340	0.018341
OPOP1120	-0.229978	0.028076
OPOP1220	0.291548	0.023988
IPIP1110	0.033601	-0.004448
IPIP1210	-0.089826	-0.003365

A0 = -0.01563 A1C = 0.00791 A1S = -0.06258

FUSELAGE PITCH ANGLE = -0.0152103

AERODYNAMIC FORCES ARE:

TORQ	XFOR	YFOR	ZFOR	HP
-85443.94	-144.40	-175.40	8120.59	-439.25

MOMX	MOMY
1323.67	3766.89

INDUCED VEL = -0.02012

VEL W.R.T. ROTOR SYSTEM ARE:
-1367.9297 0.0000 13.8721

VEL W.R.T. FUSE SYSTEM ARE:
-1367.9297 0.0000 13.8721

UNBALANCE FORCES ARE:

FX	FZ	MX	MY
-0.20	55.95	1284.19	310.01

FUSLAGE LIFT,DRAG,MOM,FX,FY,FZ:
-2.51 223.91 -4780.09 223.87 0.00 -4.78

WING LIFT,DRAG,MOM,FX,FY,FZ:
270.91 24.47 -382.35 27.22 0.00 270.65

HTAIL LIFT,DRAG,MOM,FX,FY,FZ:
-28.87 5.16 0.00 4.87 0.00 -28.92

VTAIL LIFT,DRAG,MOM,FX,FY,FZ:
56.39 12.82 0.00 12.82 56.39 0.00

NO. OF ITERATION = 6

INITIAL CONDITIONS ARE:

	VELOCITY	DISPLACEMENT
TOR 1110	-1.777967	0.010964
TOR 1120	-0.010343	0.005201
TOR 1210	1.693171	-0.069420
TOR 1220	-0.010702	-0.003582
TEET 0	0.280621	0.018438
OPOP1120	-0.230223	0.028071
OPOP1220	0.290204	0.023965

IPIP1110 0.033896 -0.004441

IPIP1210 -0.089638 -0.003350

A0 = -0.01502 A1C = 0.00826 A1S = -0.06279

FUSELAGE PITCH ANGLE = -0.0151885

AERODYNAMIC FORCES ARE:

TORQ	XFOR	YFOR	ZFOR	HP
-84954.25	-143.54	-173.00	7979.02	-436.73

MOMX	MOMY
-1854.73	3362.71

INDUCED VEL = -0.02047

VEL W.R.T. ROTOR SYSTEM ARE:
-1367.9287 0.0000 13.9675

VEL W.R.T. FUSE SYSTEM ARE:

-1367.9287 0.0000 13.9675

UNBALANCE FORCES ARE:

FX	FZ	MX	MY
0.10	-85.69	-1894.23	-110.23

FUSLAGE LIFT,DRAG,MOM,FX,FY,FZ:
-2.53 223.90 -4782.71 223.87 0.00 -4.82

WING LIFT,DRAG,MOM,FX,FY,FZ:
270.91 24.47 -382.35 27.24 0.00 270.64

HTAIL LIFT,DRAG,MOM,FX,FY,FZ:
-28.90 5.16 0.00 4.87 0.00 -28.95

VTAIL LIFT,DRAG,MOM,FX,FY,FZ:
56.39 12.82 0.00 12.82 56.39 0.00

NO. OF ITERATION = 7

INITIAL CONDITIONS ARE:

	VELOCITY	DISPLACEMENT
TOR 1110	-1.774053	0.009831
TOR 1120	-0.015659	0.005187
TOR 1210	1.686185	-0.069426
TOR 1220	-0.007411	-0.003506
TEET 0	0.281252	0.018313
OPOP1120	-0.229984	0.028074
OPOP1220	0.291828	0.023984
IPIP1110	0.033634	-0.004450
IPIP1210	-0.089879	-0.003366

A0 = -0.01565 A1C = 0.00793 A1S = -0.06254

FUSELAGE PITCH ANGLE = -0.0152208

AERODYNAMIC FORCES ARE:

TORQ	XFOR	YFOR	ZFOR	HP
-85704.56	-143.95	-177.63	8153.75	-440.59

MOMX	MOMY
1993.19	3508.09

INDUCED VEL = -0.02010

VEL W.R.T. ROTOR SYSTEM ARE:
-1367.9290 0.0000 13.9376

VEL W.R.T. FUSE SYSTEM ARE:
-1367.9290 0.0000 13.9376

```
UNBALANCE FORCES ARE:

        FX            FZ            MX            MY
       -0.14         89.10       1953.68        108.48

FUSLAGE LIFT,DRAG,MOM,FX,FY,FZ:
     -2.53     223.90  -4781.88      223.87        0.00       -4.81

WING LIFT,DRAG,MOM,FX,FY,FZ:
    270.91      24.47   -382.35       27.23        0.00      270.64

HTAIL LIFT,DRAG,MOM,FX,FY,FZ:
    -28.86       5.16      0.00        4.87        0.00      -28.91

VTAIL LIFT,DRAG,MOM,FX,FY,FZ:
     56.39      12.82      0.00       12.82       56.39        0.00

JOB TERMINATED AT NITERA=    8
**********************************************************************
COMMAND
```